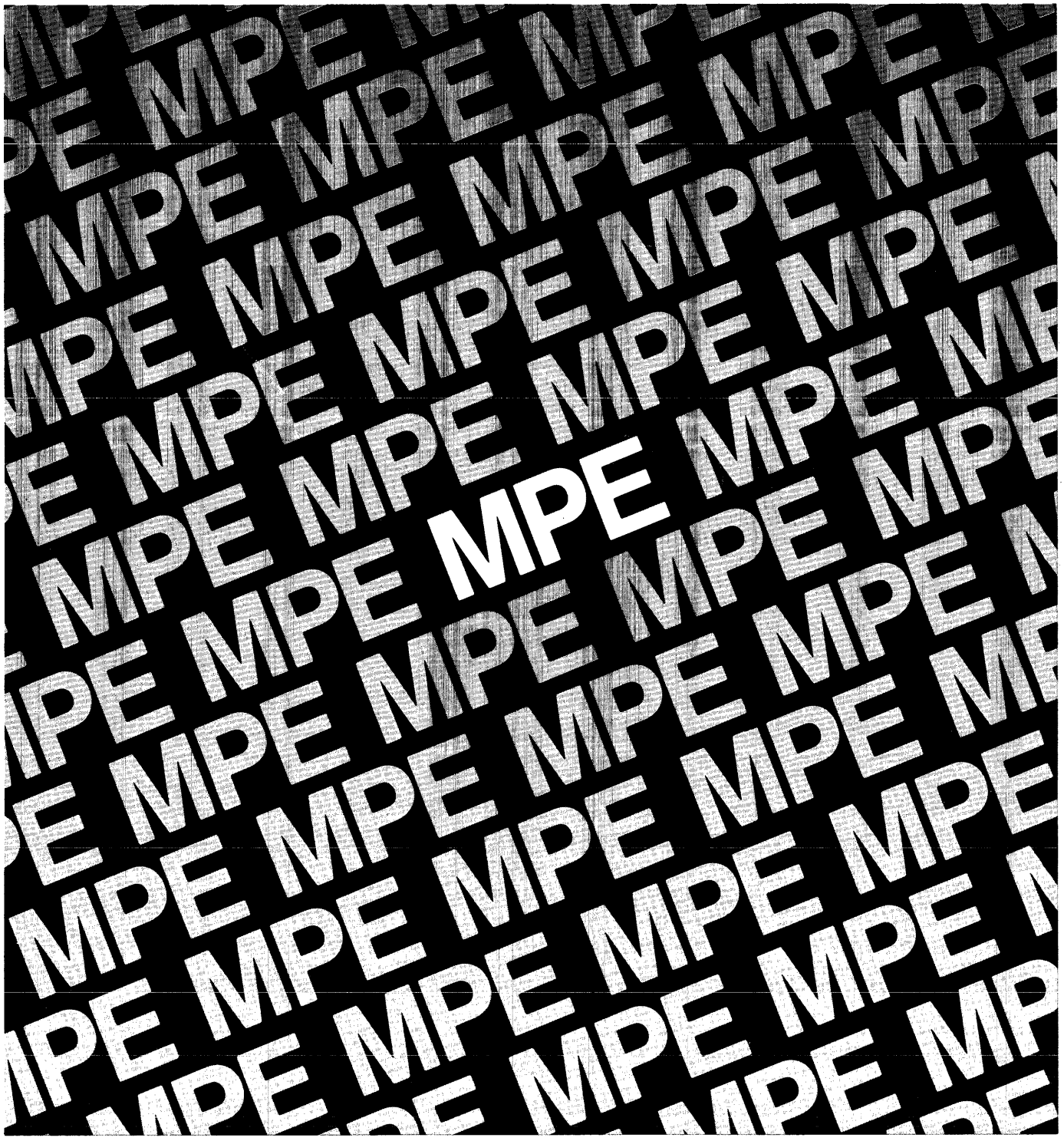
HP 3000 Computer Systems

HEWLETT PACKARD

MPE IV System Manager/System Supervisor reference manual



HP 3000 Computer Systems

MPE IV System Manager/ System Supervisor

Reference Manual

HEWLETT PACKARD

19447 PRUNERIDGE AVE., CUPERTINO, CALIFORNIA 95014

Part No. 30000-90014
Product No. 32002C
E1281

Printed in U.S.A. 12/81

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.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the date of the current edition, and lists the dates of all updated pages. Unchanged pages are listed as "ORIGINAL". Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars and dates remain. No information is incorporated into a reprinting unless it appears as a prior update.

Fifth Edition.....December 1981

Effective Pages	Date
i to viii	DEC 1982
ix to xi	ORIGINAL
xii	MAR 1982
2-3 to 2-6	DEC 1982
2-7	ORIGINAL
2-8	DEC 1982
2-9	ORIGINAL
2-10	DEC 1982
2-11 to 2-14	ORIGINAL
2-15 to 2-16	DEC 1982
2-17 to 2-25	ORIGINAL
2-26 to 2-28	DEC 1982
2-29	ORIGINAL
2-30 to 2-31	DEC 1982
2-32 to 2-35	ORIGINAL
2-36	MAR 1982
2-37 to 2-42	ORIGINAL
2-43	DEC 1982
2-44 to 2-45	ORIGINAL
2-46 to 2-46a/b	DEC 1982
2-47 to 2-57	ORIGINAL
2-58 to 2-61b	DEC 1982
2-62 to 2-70	ORIGINAL
3-1 to 3-16	ORIGINAL
4-1 to 4-16	ORIGINAL
5-1 to 5-4	DEC 1982
5-5 to 5-10	ORIGINAL
6-1 to 6-9	ORIGINAL
6-10	DEC 1982
6-11 to 6-15	ORIGINAL
6-16 to 6-18	DEC 1982
6-19 to 6-22	ORIGINAL
6-23	MAR 1982
6-24	DEC 1982
6-25 to 6-43	ORIGINAL
6-44 to 6-44a/b	DEC 1982
6-45	ORIGINAL
6-46	DEC 1982
6-47 to 6-51	ORIGINAL
6-52	DEC 1982
6-53 to 6-57	ORIGINAL
6-58	DEC 1982
6-59 to 6-69	ORIGINAL
6-70	MAR 1982
6-71 to 6-86	ORIGINAL
6-87	DEC 1982
6-88 to 6-95	ORIGINAL
6-96	MAR 1982
6-97 to 6-102	ORIGINAL
6-103 to 6-104	DEC 1982
6-105 to 6-130	ORIGINAL
6-131	DEC 1982
6-132 to 6-134	ORIGINAL
6-135 to 6-136	DEC 1982
6-137 to 6-145	ORIGINAL
6-146 to 6-147a/b	DEC 1982
6-148 to 6-162	ORIGINAL
A-1 to A-7	DEC 1982
B-1 to B-2	ORIGINAL
B-3 to B-4	DEC 1982
B-5	MAR 1982
B-6 to B-8	ORIGINAL

LIST OF EFFECTIVE PAGES (Continued)

Effective Pages	Date
B-8a/b to B-9	DEC 1982
B-10 to B-12	ORIGINAL
B-13 to B-14	DEC 1982
B-15 to B-18	ORIGINAL
C-1 to C-6	ORIGINAL
C-7	DEC 1982
D-1 to D-11	ORIGINAL
E-1 to E-2	ORIGINAL
F-1 to F-9	ORIGINAL
G-1 to G-4	ORIGINAL
H-1 to H-4	ORIGINAL
I-1	ORIGINAL
I-2	MAR 1982
I-3 to I-26	ORIGINAL
I-27 to I-28	MAR 1982
I-29 to I-34	ORIGINAL
I-35	MAR 1982
I-36 to I-38	ORIGINAL
J-1	ORIGINAL

PRINTING HISTORY

New editions are complete revisions of the manual. 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 of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

The software date code number printed alongside the date indicates the version level of the software product at the time the manual edition or update was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

First Edition	JUN 1976	32002A.00
Updates Nos. 1 and 2 Incorp	FEB 1977	32002A.00
Update No.3	MAR 1977	32002A.01
Second Edition	APR 1978	32002B.00.00
Update No.1	NOV 1978	32002B.00.01 32033A.00.00 and 32033B.00.00
Third Edition	JUL 1979	32002B.01.00 32033A.01.00 and 32033B.01.00
Update #1	DEC 1979	32033B.01DC and 32002B.01DC
Update #2	MAR 1980	32002B.01.01 and 32033B.01.01
Update #2 Incorporated	JUN 1980	32002B.01.01 and 32033B.01.01
Fourth Edition	FEB 1981	32002C
Update #1	MAY 1981	32002C
Update #2	NOV 1981	32002C
Fifth Edition	DEC 1981	32002C
Update #1	MAR 1982	32002C
Update #2	DEC 1982	C/D.01.00

PREFACE

The fifth edition of the System Manager/System Supervisor Reference Manual is intended for the HP 3000 Computer System users assigned the System Manager, Account Manager, or System Supervisor capabilities under the MPE IV operating system. The major highlights of this new edition are the HP 3000 Series 40, 40SX and 64, the 7911, 7912 and 7933 disc drives, and the integrated cartridge tape unit.

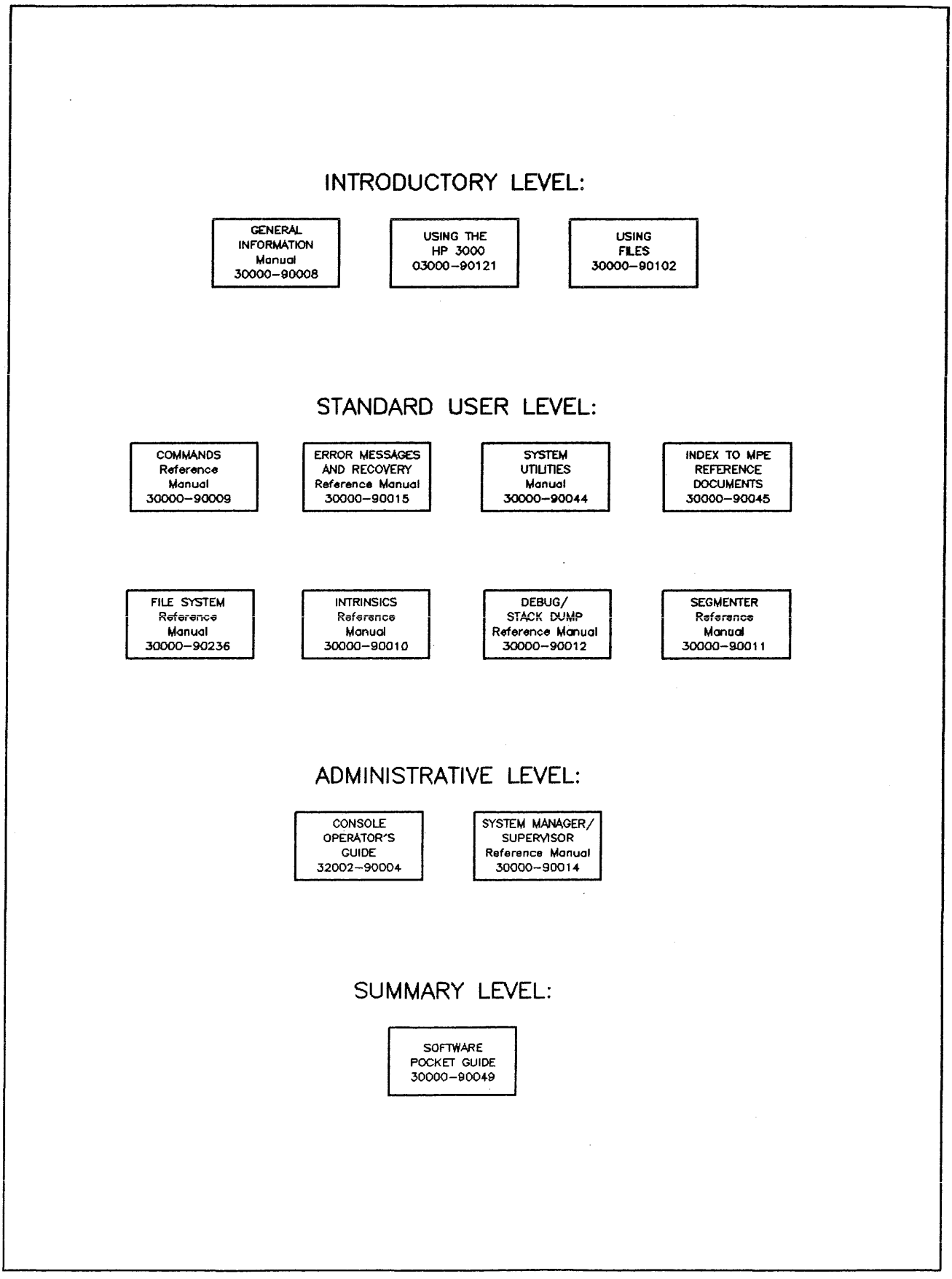
Note that unless otherwise specified, all references made to the HP 3000 Series 44 throughout this manual apply equally to the HP 3000 Series 40 System.

The user of this reference book should understand the basic operating principles of HP 3000 computer systems. The MPE Operating System, General Information Manual (Part No. 30000-90008) will give you an overview of the interrelationship between the main hardware and software features offered.

Other manuals which are available for reference are:

- *MPE Intrinsics Reference Manual (30000-90010)*
- *Using Files (30000-90102)*
- *Console Operator's Guide (32002-90004)*
- *MPE Commands Reference Manual (30000-90009)*

MANUAL PLAN


INTRODUCTORY LEVEL:
GENERAL INFORMATION Manual 30000-90008
USING THE HP 3000 03000-90121
USING FILES 30000-90102
STANDARD USER LEVEL:
COMMANDS Reference Manual 30000-90009
ERROR MESSAGES AND RECOVERY Reference Manual 30000-90015
SYSTEM UTILITIES Manual 30000-90044
INDEX TO MPE REFERENCE DOCUMENTS 30000-90045
FILE SYSTEM Reference Manual 30000-90236
INTRINSICS Reference Manual 30000-90010
DEBUG/ STACK DUMP Reference Manual 30000-90012
SEGMENTER Reference Manual 30000-90011
ADMINISTRATIVE LEVEL:
CONSOLE OPERATOR'S GUIDE 32002-90004
SYSTEM MANAGER/ SUPERVISOR Reference Manual 30000-90014
SUMMARY LEVEL:
SOFTWARE POCKET GUIDE 30000-90049

CONVENTIONS USED IN THIS MANUAL

NOTATION	DESCRIPTION
COMMAND	Commands are shown in CAPITAL LETTERS. The names must contain no blanks and be delimited by a non-alphabetic character (usually a blank).
KEYWORDS	Literal keywords, which are entered optionally but exactly as specified, appear in CAPITAL LETTERS.
parameter	Required parameters, for which you must substitute a value, appear in ***bold italics***.
parameter	Optional parameters, for which you may substitute a value, appear in *standard italics*.
[ ]	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: [A] [B] user may select A or B or neither. When brackets are nested, parameters in inner brackets can only be specified if parameters in outer brackets or comma place-holders are specified. Example: [*parm1*[,*parm2*[,*parm3*]]] may be entered as *parm1*,*parm2*,*parm3* or *parm1*,,*parm3* or ,,*parm3* ,etc.
{ }	When several elements are stacked within braces the user ***must*** select one of these elements. Example: { A } { B } user must select A or B or C. { C }
...	An ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted.
user input	In examples of interactive dialog, user input is underlined. Example: NEW NAME? ALPHA1
superscript^{c}	Control characters are indicated by a superscript^{c}. Example: Y^{c}. (Press Y and the CNTL key simultaneously.)
RETURN	RETURN indicates the carriage return key.

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

ILLUSTRATIONS

TABLES

INTRODUCTION TO COMMANDS

SECTION I

MPE commands allow you to initiate, control, and terminate the processing of programs and to request various other system functions. You generally use commands for functions external to the source-language programs that you write, although many of these functions may be necessary to support those programs. For example, with the appropriate capabilities, you would use commands discussed in this manual for:

- Obtaining accounting information for accounts and groups (:REPORT command).
- Adding or removing users from the system list of legitimate users (:NEWUSER and :PURGEUSER commands).
- Copying the software operating system (MPE) onto magnetic tape or disc for backup and modification purposes (:SYSDUMP command).
- Controlling timeslice and priorities for executing processes (:TUNE command).
- Create, alter, or remove private disc volume sets (:NEWVSET, :ALTVSET, and :PURGEVSET commands).

Other commands and functions are available as well; complete specifications for all commands are presented in Section II.

HOW TO ENTER COMMANDS

You can enter commands through any standard input device, typically the terminal for sessions or the card reader for batch jobs. Each command is accepted by the *MPE Command Interpreter*, which passes it to the appropriate system procedure for execution. Following this execution, control returns to the Command Interpreter, which now is ready for another command.

COMMAND ELEMENTS

Each MPE command consists of:

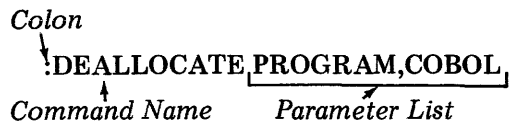
- A *colon* (required in all cases as an MPE command identifier).
- A *command name* (required in all cases).
- A *parameter list* (used in most cases).

A summary of command elements and conventions used in this manual is presented in table 1-1.

Table 1-1. Command Elements and Conventions

Leading Colon:	The prompt/command identifier character in interactive sessions. The command identifier character in batch jobs.
Command Name:	Shown in CAPITAL LETTERS IN REGULAR (ROMAN) TYPE, contains no blanks, is delimited by a non-alphanumeric character (usually a blank).
Parameters:	Shown in CAPITAL LETTERS IN REGULAR TYPE when they are literal information that you always enter exactly as shown. Shown in lower-case *italics* when they are variable parameters to be replaced by information that you must supply.
Positional Parameters:	Have significance implied by their positional order after command name; use adjacent commas (or semicolons where required) to indicate omitted parameter(s) as follows: :COMMANDNAME p1,,p3 *(from middle of list)* :COMMANDNAME ,p2,p3 *(from start of list)* :COMMANDNAME p1 *(from end of list)*
Keyword Parameters:	Separated by semicolons and can appear in any order.
Mixed Parameters:	Positional parameters are given first; first keyword indicates end of positional list.

A typical command including all three elements is as follows:



The *colon* identifies a statement as an MPE command. In a session, MPE prints the colon on the terminal whenever it is ready to accept a command. In a batch job, you must enter the colon, placing it in the first column of the source card or record on which the command is to appear.

The *command name*, which you enter immediately after the colon, requests a specific operation. MPE prohibits embedded blanks within the name, and rejects the command if they appear. MPE interprets the first non-alphanumeric character encountered as denoting the end of the command name; typically this character will be a blank.

The *parameter list* contains one or more parameters that specify operands for the command. It is required in some commands, but is optional or prohibited in others.

Normally, you must separate the parameter list from the command name by one or more blanks as *delimiters*. Within the list, any delimiter can be surrounded by any number of blanks, permitting a free and flexible command format.

MPE permits both decimal and octal numbers as command parameters. You can distinguish the two by preceding the octal number with a percent sign (%).

The end of each command is indicated by the end of the record on which it appears — for example, a *carriage return* for terminal input or the *end of the card* containing the command for card input. If the last non-blank character of the record is an ampersand (&), this signifies a continuation character and the command is continued onto the next record.

NOTE

When you are running programs in batch job mode, bear in mind that MPE scans the entire record of each card image; thus no characters are ignored.

KEYWORD PARAMETERS

The meaning of a keyword parameter group is independent of its position in the list of parameters; thus you can enter keyword groups in any order with respect to each other. A *keyword group* consists of a keyword that denotes its meaning, optionally followed by an equal sign and one or more sub-parameters. Each keyword group is preceded by a semicolon. When more than one sub-parameter appears in a group, they are separated from each other by commas. All delimiters can optionally be preceded or followed by blanks. The following example shows a :STORE command containing a keyword parameter (SHOW).

```
:STORE PROG1,PROG2;*MYTAPE;SHOW
```

CONTINUATION CHARACTERS

When the length of a command exceeds one record, you may enter an ampersand (&) as the last non-blank character of this record and continue the command on the next record. In this case, the next record must begin with a colon (supplied automatically by MPE in a session but entered by you in a batch job).

You can continue commands for up to 268 characters, including prompting colons and continuation ampersands. In continuing a command onto another line, you must not divide any word.

MPE does not begin interpretation of a command until the last record of the command is read. It then joins all records comprising the command and replaces all prompt characters and continuation ampersands with blanks.

COMMAND ERRORS

If you make an error while entering a command, MPE informs you of this through a message printed on your standard list device. You then can re-enter the command or correct it with the MPE :REDO command (see the *MPE Commands Reference Manual*). Messages and their meanings are described in the *MPE Error Messages Manual.*

EXECUTING COMMANDS PROGRAMMATICALLY

In addition to entering commands directly through your standard list device, MPE allows you to execute many of them from within programs. You do this by calls to the COMMAND intrinsic. This intrinsic calls the Command Interpreter and passes the command images to it so that MPE can interpret and execute the command. The command specifications in Section II point out those commands which can be executed from a program. See the *MPE Intrinsics Reference Manual* for a discussion of the COMMAND intrinsic.

COMMAND DEFINITIONS

SECTION II

The reference specifications for all MPE commands available to System Managers, Account Managers, and System Supervisors appear in this section. For easy reference, they are presented alphabetically by command name. For each command, the reference specifications show the following information:

- Syntax.
- Parameter definitions (including meaning, constraints, defaults).
- When legal (in jobs, sessions, during break, programmatically).
- Whether interruptable (with BREAK key).
- Capability required (System Supervisor (OP), System Manager (SM), or Account Manager (AM)).
- Operation (description of how to use command).
- Example.
- Where described in text portion of manual.

In the reference specifications, the indication *Available In Break* means: after pressing the BREAK key on the terminal (or calling the CAUSEBREAK intrinsic in a program), but before entering :RESUME or :ABORT. The notation *Available Programmatically* means: using the COMMAND intrinsic. (The CAUSEBREAK and COMMAND intrinsics are explained in the *MPE Intrinsics Reference Manual.*)

The capabilities required are defined as follows:

OP	System Supervisor
SM	System Manager
AM	Account Manager

The commands are grouped by function in table 2-1.

Table 2-1. Functional List of Commands

FUNCTION	COMMAND
Program/Procedure Allocation	:ALLOCATE :DEALLOCATE
Account Management	:NEWACCT :NEWGROUP :NEWUSER :ALTACCT :ALTGROUP :ALTUSER :LISTACCT :LISTGROUP :LISTUSER :PURGEACCT :PURGEGROUP :PURGEUSER :RESETACCT
Managing Files	:NEWVSET :ALTVSET :PURGEVSET :STORE :RESTORE
Job/session and System Performance Management	:JOBPRI :TUNE
System Reports and Information Management	:REPORT :RESUMELOG :SHOWLOG :SHOWQ :SWITCHLOG
System Modification and Back-Up	:SYSDUMP

:ALLOCATE

EXAMPLE

To permanently allocate a procedure identified as PROC1, residing in SL.PUB.SYS, enter:

```
:ALLOCATE PROCEDURE,PROC1
```

NOTE

Program files residing in the non-system domain (private volume) will not be allocated. Attempts to do so will result in LOAD ERR 92.

TEXT DISCUSSION

Page 6-129

:ALLOCATE

Loads a program or procedure.

SYNTAX

```
:ALLOCATE [PROCEDURE,] name
          [PROGRAM,   ]
```

PARAMETERS

PROGRAM	The program file indicated by *name* is allocated.
PROCEDURE	The procedure indicated by *name* in SL.PUB.SYS is allocated. The default is PROGRAM.
name	The name of the program file or procedure to be allocated.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	NO
Breakable?		NO
Capability Required:		OP

OPERATION

This command loads the code segments for a program or procedure. These segments remain loaded until DEALLOCATED. The program file specified in the :ALLOCATE command must be a permanent file. LOAD ERR 46 will result if a temporary program file is specified.

The user issuing the :ALLOCATE command must have EXECUTE access for any file referenced in the *name* parameter of this command.

Any external procedures referenced by a program being allocated by this command must reside in SL.PUB.SYS.

:ALTACCT

Changes the attributes of an existing account.

SYNTAX

```
:ALTACCT acctname
              [;PASS=[password]]
              [;FILES=[filespace]]
              [;CPU=[cpu]]
              [;CONNECT=[connect]]
              [;CAP=[capabilitylist]]
              [;ACCESS=[(fileaccess)]]
              [;MAXPRI=[subqueuename]]
              [;LOCATTR=[localattribute]]

              [;VS=volset:{ALT }     ]
                          {SPAN}
```

PARAMETERS

acctname — The name of the account to be altered.

password — Account password (used for verifying log-on access only). If omitted, any existing password is removed.

filespace — Disc storage limit, in sectors, for the permanent files in the account. *Filespace* limit cannot be less than the number of sectors currently in use for the account. Default is unlimited filespace.

cpu — The limit on cpu time, in seconds, for the account. This limit is checked only when a job/session is initiated - thus, the limit never causes a job/session to abort. The maximum value permitted is 2,147,483,647 seconds. Default is unlimited cpu time.

connect — The limit on total session connect time, in minutes, allowed the account. This limit is checked only at log on. The maximum value allowed is 2,147,483,647 minutes. Default is unlimited connect time.

capabilitylist — List of capabilities, mutually separated by commas, permitted this account. Each capability is denoted by a two-letter mnemonic, as follows:

System Manager	=	SM
Account Manager	=	AM
Account Librarian	=	AL
Group Librarian	=	GL
Diagnostician	=	DI
System Supervisor	=	OP
Permanent Files	=	SF
Access of non-sharable I/O devices	=	ND

Use Volumes	=	UV
Create Volumes	=	CV
Use Communications Subsystem	=	CS
User Logging	=	LG
Process Handling	=	PH
Extra Data Segments	=	DS
Multiple RINS	=	MR
Privileged Mode	=	PM
Interactive Access	=	IA
Local Batch Access	=	BA

Default is AM, AL, GL, SF, ND, IA, BA. Note that CV automatically gives the account UV capability.

fileaccess

File security specifications, entered as follows:

```
          [R]
          [L]
        [[A][,...] :{ANY} ][;...]
          [W]        {AC }
          [X]
```

where R, L, A, W and/or X specify modes of access by types of users (ANY and/or AC) as follows:

R = Read
L = Lock (allows exclusive access to file)
A = Append (implicitly specifies L also)
W = Write (implicitly specifies A and L also)
X = Execute

Two or more modes may be specified if they are separated by commas.

The user types are specified as follows:

ANY = Any user
AC = Member of this account only

Two or more user types may be specified if they are separated by commas.

The default is no security restrictions at the account level.

subqueuename

Name of the subqueue of highest priority that can be requested by any process of any job/session in the account, specified as xS, where x is A, B, C, D, or E. Default is CS.

:ALTACCT

CAUTION

Processes capable of executing in the "AS" or "BS" classes can deadlock the system. System processes and other user processes can be completely locked out from execution by "AS" and "BS" priority processes. Care should be used in assigning these priorities.

localattribute — Local attribute of the account, as defined at the installation site. This is a double-word bit map of arbitrary meaning which might be used to further classify accounts. While it is not involved in standard MPE security provisions, it is available to processes through the WHO intrinsic for use in the programmer's own security provisions. Default is double-word 0 (null).

volset:SPAN — *Volset* is a volume set or class reference which, when fully qualified, is in the form *vcsid .groupname .acctname* where *vcsid* refers to a previously-defined volume set or class definition.

SPAN indicates that the *acctname* is to be inserted in the accounting directory of the specified volume set (*volset*). The specified volume set must have been previously mounted (via a MOUNT command) for the SPAN operation to succeed.

volset:ALT — ALT directs the altering of an account or group entry on the specified volume set. ALT is useful only if it is necessary to alter account and group file space limits for entries that have already been spanned.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capability required:		SM

OPERATION

This command changes the attributes of an existing account. When altering the *capability list* for the SYS account, the SM capability cannot be removed. If *acctname* is SYS, and the file-access parameter is omitted, the following default security is assigned:

```
(R, X: ANY; A, W, L: AC)
```

When an entire keyword parameter group is omitted from the :ALTACCT command, that parameter remains unchanged for the account. When a keyword is included but the corresponding parameter is omitted (as in ;PASS=), the default value is assigned as follows:

Parameter	Default Values
password	No password
filespace	Unlimited
cpu	Unlimited
connect	Unlimited
capabilitylist	*AM*, *AL*, *GL*, SF, ND, IA, BA (All accounts except SYS)
	SM, AM, AL, GL, DI, OP, SF, ND, PH, DS, MR, PM, IA, BA (SYS account only)
fileaccess	R, A, W,L, X: AC (All accounts except SYS)
	R, X: ANY; A, W, L: AC (SYS account only)
subqueuename	CS subqueue
localattribute	Double-word 0 (null)

When changing one capability in a *capabilitylist* presently containing several non-default values, you must re-specify the entire new *capabilitylist*, not just the changes parameter. Also, users within the account who have the non-default capabilities may have to be modified.

Any value changed through the :ALTACCT command becomes effective the next time MPE is requested to check this value; if an attribute is taken away from an account, users of this account currently running with that attribute will retain it until again logging on. For this reason, it is recommended that all account users be advised of all changes in advance.

EXAMPLE

To change an account named AC2 so that its *password* is GLOBALX and its *filespace* is limited to 50,000 sectors, enter:

```
:ALTACCT AC2;PASS=GLOBALX;FILES=50000
```

TEXT DISCUSSION

Page 4-5.

:ALTGROUP

Changes one or more attributes of a group.

SYNTAX

```
:ALTGROUP groupname
          [;PASS=[password]]
          [;CAP=[capabilitylist]]
          [;FILES=[filespace]]
          [;CPU=[cpu]]
          [;CONNECT=[connect]]
          [;ACCESS=[(fileaccess)]]
          [;VS=volset:{ALT }  ]
                      {SPAN}
```

PARAMETERS

groupname — Specifies the name of the group whose attributes are to be changed.

password — The new password to be assigned to the group. (This password is used to verify log-on access only.) If omitted, any existing password is removed.

capabilitylist — Specifies the list of capabilities permitted this group. Each capability is denoted by a two-letter mnemonic, as follows:

Process Handling	=	PH
Extra Data Segments	=	DS
Multiple RINs	=	MR
Privileged Mode	=	PM
Interactive Access	=	IA
Local Batch Access	=	BA

Two or more capabilities may be specified if they are separated by commas.

Default is IA, BA.

filespace — Disc storage limit, in sectors, for the permanent files of the group. A group's *filespace* cannot be specified as greater than the corresponding limits currently defined for the group's account. However, an account's *filespace* can later be changed so that some of its groups are left with limits that exceed the new account limits.

Default is unlimited file space.

cpu

The limit on total cpu time, in seconds, for the group. This limit is checked only when a job or session is initiated — thus, the limit never causes a job/session to abort. The maximum value permitted is 2,147,483,647 seconds. A group's cpu limit cannot be specified as greater than the corresponding limit currently defined for the group's account. However, an account's cpu limit can be changed later so that some of its groups are left with a limit that exceeds the new account limits. Default is unlimited cpu time. If the limit is exceeded, users with Account Manager capability will be warned when logging on; other users will be denied access.

connect

The limit on total session connect time, in minutes, allowed the group. This limit is checked only at logon. The maximum allowed value is 2,147,483,647 minutes. The default is that no limit is assigned. A group's connect limit cannot be specified as greater than the corresponding limit currently defined for the group's account. However, an account's connect limit can be changed later so that some of its groups are left with a limit that exceeds the new account limits. Default is unlimited connect time. If the limit is exceeded, users with Account Manager capability will be warned when logging on; other users will be denied access.

fileaccess

File security specifications, entered as follows:

$$\left\{\begin{matrix}R\\L\\A\\W\\X\\S\end{matrix}\right\}:\left\{\begin{matrix}ANY\\AC\\GU\\AL\\GL\end{matrix}\right\}$$

where R, L, A, W, X, S specify modes of access by types of users (Any, AC, GU, AL, GL) as follows:

R = Read
L = Lock (allows exclusive access to file)
A = Append (implicitly specifies L also)
W = Write (implicitly specifies A and L also)
X = Execute
S = Save

Two or more modes may be specified if they are separated by commas.

The user types are specified as follows:

ANY = Any user
AC = Member of this account only
GU = Member of this group only
AL = Account librarian user only
GL = Group librarian user only

Two or more user types may be specified if they are separated by commas.

Default is R, A, W, L, X, S : GU (all groups except PUB); or R, X : ANY; A, W, L, S : AL, GU (PUB group only).

:ALTGROUP

volset:SPAN

Volset is a volume set or class reference which, when fully qualified, is in the form

vcsid.groupname.acctname

where *vcsid* refers to a previously-defined volume set or class definition.

If *volset* is different from the old volume set, and the old volume set is the system set, then the old volume set must be examined for the presence of files belonging to *groupname*. If no such files are found, the command will succeed, and the group is reassigned to *volset* as its new home volume set.

If *volset* is omitted from the VS= parameter, the group will be reassigned to the system volume set. If the old volume set already is the system volume set, nothing will be altered.

It is permissible to reassign a group to a different volume set regardless of the presence of files belonging to *groupname*, so long as the old volume set is not the system volume set and the *groupname* is not currently bound (either explicitly via the MOUNT command or implicitly via the FOPEN intrinsic) to its home volume set.

SPAN indicates that *groupname* is to be inserted in the accounting directory of the specified volume set (*volset*). The specified volume set must have been previously mounted (via the MOUNT command) for the span operation to succeed.

volset:ALT

ALT directs the altering of an account or group entry on the specified volume set. ALT is required to alter account and group file space limits for entries that have already been spanned. ALT is also required for changing security specifications on a volume set.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capability Required?		AM

OPERATION

This command changes one or more attributes of a group. When an entire parameter is omitted from an :ALTGROUP command, the corresponding value for the group remains unchanged. When a keyword is included but the corresponding parameter is omitted (as in ;PASS=), the default value is assigned as follows:

Parameter	Default
password	Null (no password)
capabilitylist	IA, BA (except PUB.SYS)
filespace	Unlimited
cpu	Unlimited
connect	Unlimited
fileaccess	R, A, W, L, X, S: GU (All groups except PUB) R, X: ANY; A, W, L, S: AL, G, U (PUB group only)

When a parameter is modified through :ALTGROUP, it immediately takes effect in the directory but does not apply to current accesses under the group; thus, if an attribute is taken away from a group, users of the group currently running with that attribute retain it until again logging on. For this reason, it is recommended that all group users be advised of the change in advance.

EXAMPLE

To assign a new password of PASS2 to a group named GROUPX, enter:

```
:ALTGROUP GROUPX;PASS=PASS2
```

TEXT DISCUSSION

Page 5-2.

:ALTUSER

Alters attributes currently defined for a user.

SYNTAX

```
:ALTUSER username
    [;PASS = [password]]
    [;CAP = [capabilitylist]]
    [;MAXPRI = [subqueuename]]
    [;LOCATTR = [localattribute]]
    [;HOME = [homegroupname]]
```

PARAMETERS

username The name assigned to the user (with the :NEWUSER command or the first user assigned with a :NEWACCT command). (REQUIRED PARAMETER).

password The new password to be assigned to the user. If omitted, any existing password is removed.

capabilitylist List of capabilities, mutually separated by commas, permitted this account. Each capability is denoted by a two-letter mnemonic, as follows:

System Manager	=	SM
Account Manager	=	AM
Account Librarian	=	AL
Group Librarian	=	GL
Diagnostician	=	DI
System Supervisor	=	OP
Permanent Files	=	SF
Access of non-sharable I/O devices	=	ND
Use Volumes	=	UV
Create Volumes	=	CV
User Communications Subsystem	=	CS
User Logging	=	LG
Process Handling	=	PH
Extra Data Segments	=	DS
Multiple RINS	=	MR
Privileged Mode	=	PM
Interactive Access	=	IA
Local Batch Access	=	BA

Default is SF, ND, IA, BA. Note that CV automatically gives the user UV capability.

subqueuename The name of the subqueue of highest priority that can be requested by any process of any job/session in the account, specified as xS, where x is A, B, C, D, or E.

The priority specified with the :ALTUSER command cannot be greater than that specified with the :NEWACCT or :ALTACCT commands. (But, if the user's account is altered later, he may be left with excessive limits.) The *subqueuename* defined for the user is checked against the *subqueuename* defined for his account at log-on time, and the lower priority of the two is used as the maximum priority restricting all processes of the job/session. Also, the priority requested by the user when he logs on is checked against the *subqueuename* defined for him, and he is granted the lower of these two values. Default is CS.

CAUTION

Processes capable of executing in the 'AS' or 'BS' classes can deadlock the system. System processes and other user processes can be completely locked out from execution by 'AS' and 'BS' priority processes. Care should be used in assigning these priorities.

localattribute The local attribute of the user, as defined at the installation site. This is a double-word bit map of arbitrary meaning which might be used to further classify users. When it is not involved in standard MPE security provisions, it is available to processes through the WHO intrinsic for use in the programmer's own security provisions. The bit map for the user local attributes must be a subset of the bit map for the account local attributes. The :ALTUSER command checks the user's local attributes with those of the account's. Default is double-word 0 (null).

homegroupname The name of an existing group to be assigned as the homegroup for this user. Default is no homegroup.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required?		AM

OPERATION

This command alters the attributes currently defined for a user.

When an entire keyword parameter group is omitted from the :ALTUSER command, the corresponding value for this user remains unchanged. When a keyword is included but the corresponding parameter is omitted (as in ;PASS=), a default value is assigned as follows:

:ALTUSER

Parameter	Default Values
password	Null (no password)
capabilitylist	SF, ND, IA, BA (provided the account has these capabilities)
subqueuename	CS
localattribute	Null (double-word 0)
homegroupname	None (no home group). If a user has no homegroup assigned, he must specify a log-on group when initiating a job or session.

When a parameter is modified with the :ALTUSER command, it takes effect immediately in the directory, but does not apply to users currently logged on; it will take effect for the next log on by such users. For this reason, it is recommended that the users involved be advised of the change in advance. It is not recommended that the *capabilitylist* or *homegroupname* of the MANAGER.SYS user be altered.

EXAMPLE

Suppose an account's capabilities are AM, AL, GL, SF, ND, PH, DS, MR, IA, BA. To change the capabilitylist of the user JONES from IA, BA, SF, PH, DS to include the Multiple RIN capability (MR), enter:

```
:ALTUSER JONES;CAP=IA,BA,SF,PH,DS,MR
```

TEXT DISCUSSION

Page 5-4.

:ALTVSET

Modifies volume set definitions.

SYNTAX

```
:ALTVSET vsname
        [;ADDCLASS= vcname:vname[,vname],...[,vname]]
        [;EXPANDCLASS= vcname:vname[,vname],...[,vname]]
        [;EXPANDSET= vname:type[,vname:type],...[vname:type]]
```

PARAMETERS

vsname — Existing volume set name. MPE will implicitly reference *vsname* as *vsname.groupname.acctname*, where *groupname* and *acctname* are the log-on group and account.

vcname:vname — Volume class name, consisting of up to eight alphanumeric characters, beginning with a letter. Each volume class definition consists of a volume class name and an associated list of volume names, *vname*. The list of volume names must be a subset of the volumes comprising the volume set. One of the volume names must be that of the volume set's master volume.

;ADDCLASS — Add a volume class and an associated list of volume names. The list of volume names must be a subset of the volumes comprising the volume set. One of the volume names must be that of the volume set's master volume.

;EXPANDCLASS — Add names of new members in the volume class. These new members can only be members of the parent volume set.

;EXPANDSET — Add names of new members in the volume set and types of devices required to accommodate the new members. Total members in a volume set may not exceed eight. The type must be specified with each volume member name to define the type of disc drive. Types of disc drives that may be specified are HP 7902, HP 7905, HP 7906, HP 7920, HP 7925, HP 7933, and HP 9895.

:ALTVSET

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capability required:		CV

OPERATION

The :ALTVSET command is used to modify volume set definitions. Volume class names can be added, and the number of member volumes can be increased. The overall size of the volume set, however, cannot be decreased (member volumes cannot be deleted). Volumes cannot be deleted from a volume set or class.

EXAMPLE

To add member volumes MEM3 and MEM4 to the existing volume set USERSET, enter:

```
:ALTVSET USERSET;EXPANDSET=MEM3:HP7920,MEM4:HP7920
```

TEXT DISCUSSION

Page 4-13.

:DEALLOCATE

Removes from virtual memory a program file or procedure segment that has been permanently allocated with an :ALLOCATE command.

SYNTAX

```
                  PROGRAM
:DEALLOCATE [                ,]name
                  PROCEDURE
```

PARAMETERS

PROGRAM The program file indicated by *name* is deallocated. Default.

PROCEDURE The code segment containing the procedure specified by *name* in SL.PUB.SYS is deallocated. Default if PROGRAM.

name The name of the program file or procedure to be deallocated. (REQUIRED PARAMETER).

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	No
	programmatically?	Yes
Breakable?		No
Capability Required:		OP

OPERATION

This command removes from virtual memory a program file or procedure segment that has been permanently allocated with an :ALLOCATE command.

The program file or procedure segment is deallocated immediately if it is not being used by a process; otherwise, it is deallocated as soon as it is no longer in use.

EXAMPLE

To deallocate a program file named PROGX, enter:

```
:DEALLOCATE PROGX
```

TEXT DISCUSSION

Page 6-129.

:IMFMGR

Starts the interactive IMF Manager's Program.

SYNTAX

:IMFMGR

PARAMETERS

None

USE

Available?	in Session?	Yes
	in Job?	Yes
	in Break?	No
	Programmatically?	No
Breakable?		Yes
Capability Required:		OP

OPERATION

This command starts the IMF Manager's Program, which makes available the IMF commands that are used to verify IMF configuration files, display the names of IMF configuration files and devices in use, start and stop the Pass Through Mode (PTM) on selected HP 3000 terminals and printers, start and stop communications with one or more host systems, and turn the CS/3000 Trace utility on and off.

After you type IMFMGR, you will be prompted with a "> " (rivet).

EXAMPLE

To initiate the IMF Manager's Program, type:

:IMFMGR

TEXT DISCUSSION

See the *IMF Manual*.

:JOBPRI

Sets or changes the default execution priority for batch jobs and sets a maximum execution priority for jobs.

SYNTAX

```
:JOBPRI [maxsubqueue] [,defaultsubqueue]
```

PARAMETERS

maxsubqueue The maximum priority at which batch jobs will be allowed to run. This is in spite of any job priority a user may have requested with the :JOB command. This parameter may be ES, DS, CS, or 0. *If 0 is specified, no limit is imposed on batch jobs.* Default is no change in maximum priority.

defaultsubqueue The default execution priority for batch jobs, which may be ES, DS, or CS. This takes effect if a user does not specify an execution priority on his :JOB command. Default is no change in execution priority.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required:		OP

OPERATION

This command changes the default execution priority for batch jobs and sets a maximum execution priority for jobs. The *maxsubqueue* parameter takes precedence over the *default-subqueue* parameter. That is, if the default parameter is greater than the maximum priority parameter, jobs will be initiated with the maximum priority parameter.

EXAMPLE

To raise the maximum execution priority so that batch jobs can run in any subqueue requested, enter:

```
:JOBPRI 0
```

TEXT DISCUSSION

None

:LISTACCT

Lists the attributes for one account (Account Manager capability) or all accounts (System Manager capability).

SYNTAX

```
:LISTACCT [acctset] [,listfile]
```

PARAMETERS

acctset Specifies the set of groups to be listed. This parameter can be entered in one of the following formats:

acctdesig The name of the account whose attributes are to be listed. Account Managers can specify only their own account.

(Required parameter for Account Managers, optional parameter for System Manager)

@ All accounts. Default. (System Manager capability required.)

NOTE

The characters @, #, and ? can be used as "wild card" characters in the acctset parameter. These wild card characters have the following meanings:

@ — specifies zero or more alphanumeric characters.

— specifies one numeric character.

? — specifies one alphanumeric character.

The characters can be used as follows:

n@	List all accounts starting with the character *n*.
@*n*	List all accounts ending with the character *n*.
n@x	List all accounts starting with the character *n* and ending with the character *x*.
n##..#	List all accounts starting with the character *n* followed by up to seven digits.
?*n*@	List all accounts whose second character is *n*.
n?	List all two-character accounts starting with the character *n*.
?*n*	List all two-character accounts ending with the character *n*.

listfile Actual designator of file on which the attribute listing is produced. This file is closed with SAVE disposition. Default is $STDLIST.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes. Aborts
Capability Required:		AM or SM

OPERATION

This command lists the attributes for one account (Account Manager capability) or all accounts (System Manager capability). Private volume information (volume set names) also is listed. The volume sets need not be mounted for this information to be listed.

EXAMPLE

To list the attributes of his own account, the Account Manager user enters:

:LISTACCT MYACCT

TEXT DISCUSSION

Page 5-9.

:LISTGROUP

Prints a list of attributes for groups.

SYNTAX

```
:LISTGROUP [groupset] [,listfile]
```

PARAMETERS

groupset Specifies the set of groups to be listed. Default is @. This parameter can be entered in one of the following formats:

groupdesig.acctdesig	Specifies the group and account. If an Account Manager specifies an account, it must be his own.
groupdesig	Specifies the group (in the log-on account).
@.acctdesig	All groups in the specified account. If an Account Manager specifies an account, it must be his own.
@	All groups in the log-on account. Default.
@.@	All groups in all accounts. Only System Managers may specify this option.

NOTE

The characters @, #, and ? can be used as "wild card" characters in the groupset parameter. These wild card characters have the following meanings:

@	—	species zero or more alphanumeric characters.
#	—	species one numeric character.
?	—	specifies one alphanumeric character.

The characters can be used as follows:

n@	List all groups starting with the character *n*.
@*n*	List all groups ending with the character *n*.
n@x	List all groups starting with the character *n* and ending with the character *x*.
n##..#	List all groups starting with the character *n* followed by up to seven digits.
?*n*@	List all groups whose second character is *n*.
n?	List all two-character groups starting with the character *n*.
?*n*	List all two-character groups ending with the character *n*.

listfile Actual designator of file on which attribute listing is produced. This file is closed with SAVE disposition. Default is $STDLIST.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes. Aborts.
Capability Required:		AM or SM

OPERATION

This command lists attributes for groups. An Account Manager may specify only groups in his own account; the System Manager may specify all groups in all accounts on the system. Private volume information (volume set names) also is listed. The volume sets need not be mounted for this information to be listed. A sample listing and explanation of the octal decoding is given in Appendix D.

EXAMPLE

To list the attributes of all groups in the account named ACCT1, enter:

```
:LISTGROUP @.ACCT1
```

TEXT DISCUSSION

Page 5-9.

:LISTUSER

Lists attributes currently assigned to users.

SYNTAX

```
:LISTUSER [userset] [,listfile]
```

PARAMETERS

userset Specifies the set of users to be listed. Default is @. This parameter is written in one of the following formats:

userdesig.acctdesig	The specified user in the specified account. If Account Manager users specify *acctdesig*, it must be their own account.
userdesig	The specified user in the log-on account.
@.acctdesig	All users in the specified account. Account Managers may specify their own account only.
@	All users in the log-on account. Default.
@ . @	All users in all accounts. (The attributes of the accounts also are listed.) This option can only be specified by System Manager.

NOTE

The characters @, #, and ? can be used as "wild card" characters in the userset parameter. These wild card characters have the following meanings:

@	—	specifies zero or more alphanumeric characters.
#	—	specifies one numeric character.
?	—	specifies one alphanumeric character.

The characters can be used as follows:

n@	List all users starting with the character *n*.
@*n*	List all users ending with the character *n*.
n@*x*	List all users starting with the character *n* and ending with the character *x*.
n##..#	List all users starting with the character *n* followed by up to seven digits.
?*n*@	List all users whose second character is *n*.
n?	List all two-character users starting with the character *n*.
?*n*	List all two-character users ending with the character *n*.

listfile Actual designator of file on which attribute listing is produced. This file is closed with SAVE dispostion. Default is $STDLIST.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes. Aborts.
Capability Required:		AM or SM

OPERATION

This command lists the attributes currently assigned to users. An Account Manager may list all users in his account only; the System Manager may list all users in all accounts. A sample listing and explanation of the octal decoding is given in Appendix D.

EXAMPLE

To list the attributes of the user named CLANCY in the account named MGT, enter:

:LISTUSER CLANCY.MGT

Note that if the command is entered by an Account Manager, MGT must be his own account.

TEXT DISCUSSION

Page 5-9.

:NEWACCT

Creates a new account and an associated Account Manager and Public Group.

SYNTAX

```
:NEWACCT acctname,mgrname
                [;PASS=[password]]
                [;FILES=[filespace]]
                [;CPU=[cpu]]
                [;CONNECT=[connect]]
                [;CAP=[capabilitylist]]
                [;ACCESS=[(fileaccess)]]
                [;MAXPRI=[subqueuename]]
                [;LOCATTR=[localattribute]]
                [;VS=volset:SPAN]
```

PARAMETERS

acctname Name to be assigned to the new account.

mgrname Name of the account manager; this is always the first user created under the account. He receives the following attributes:

User password	No password.
Capability list	Same as account's capabilities.
Scheduling priority	Same as account's maximum priority.
Local attribute.	Same as account's.
Home group	PUB.

The attributes of an Account Manager may be changed later with the :ALTUSER command. In no case, however, is this user granted effective attributes greater than those of the account.

password Account password, used for verifying log-on access only. Default is no password assigned.

filespace Disc storage limit, in sectors, for the permanent files of the account. The maximum value permitted is 2,147,483,647 sectors. Default is unlimited *filespace*.

cpu Limit on total cpu time, in seconds, for this account. This limit is checked only when a job/session is initiated--thus, the limit never causes a job/session to abort. The maximum value permitted is 2,147,483,647 seconds. Default is no limit is assigned.

connect

Limit on total session connect time, in minutes, allowed the account. This limit is checked only at log-on time. The maximum value allowed is 2,147,483,647 minutes. Default is no limit is assigned.

capabilitylist

List of capabilities, mutually separated by commas, permitted this account. Each capability is denoted by a two-letter mnemonic, as follows:

System Manager	=	SM
Account Manager	=	AM
Account Librarian	=	AL
Group Librarian	=	GL
Diagnostician	=	DI
System Supervisor	=	OP
Permanent Files	=	SF
Access of non-sharable I/O devices	=	ND
Use Volumes	=	UV
Create Volumes	=	CV
Use Communications Subsystem	=	CS
User Logging	=	LG
Process Handling	=	PH
Extra Data Segments	=	DS
Multiple RINS	=	MR
Privileged Mode	=	PM
Interactive Access	=	IA
Local Batch Access	=	BA

Default is AM, AL, GL, SF, ND, IA, BA. Note that CV automatically gives the account UV capability.

fileaccess

The restrictions of file access pertinent to this account. Default is R, A, L, W, X: AC. (See :ALTACCT command for details on access types.)

subqueuename

The name of the subqueue of highest priority that can be requested by any process of any job in the account, specified as *x*S, where *x* is A, B, C, D, or E. Default is CS.

CAUTION

Processes capable of executing the "AS" or "BS" classes can deadlock the system. System processes and other user processes can be completely locked out from execution by "AS" and "BS" priority processes. Care should be used in assigning these priorities.

localattribute

The local attribute of the account, as defined at the installation site. This is a double-word bit map of arbitrary meaning which might be used to further classisfy accounts. While it is not involved in standard MPE security provisions, it is available to processes through the WHO intrinsic for use in the programmer's own security provisions. Default is double-word 0.

volset:SPAN

Volset is a volume set or class reference which, when fully qualified, is in the form *vcsid .groupname .acctname*, where *vcsid* refers to a previously-defined volume set or class definition.

SPAN indicates that the *acctname* is to be inserted in the accounting directory of the specified volume set (*volset*). The specified volume set must have been previously mounted for the SPAN operation to succeed.

Note that the associated Account Manager and PUBLIC group are not created in the accounting directory of the specified volume set.

Once the account has been spanned to the volume set it does not need to be spanned again (i.e., the *volset* parameter is not necessary) on another system which shares the volume set under the same account name.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capability required:		SM

OPERATION

This command creates a new account and an associated Account Manager and Public Group.

Default parameters are assigned to this command as follows:

1. When a keyword is included but its corresponding keyword parameter is omitted (as in ;PASS=).

2. When an entire keyword parameter grouping (such as ;PASS=password) is omitted.

After the System Manager enters the :NEWACCT command, the new Account Manager then can redefine his own attributes, the Public Group, and new users and groups. Note that the Account Manager can only define groups and users within the set of parameters assigned by the System Manager at account creation time.

The Public Group is initially assigned no password and the same capability-class attributes, permanent file space limit, cpu time limit, and connect time limit as the account. Its initial security allows reading and program-execute access to all users who can pass the account's security, and appending, locking, and saving access to account librarian and group users only. (These access provisions are R, X: ANY; A, W, L, S: AL, GU.)

EXAMPLE

To create an account with the account name ACI and the Account Manager name SMYTHE, enter:

```
:NEWACCT ACI,SMYTH
```

All other parameters are assigned by default.

TEXT DISCUSSION

Page 4-4.

:NEWGROUP

Create a new group within an account.

SYNTAX

```
:NEWGROUP groupname
          [;PASS=[password]]
          [;FILES=[filespace]]
          [;CPU=[cpu]]
          [;CONNECT=[connect]]
          [;CAP=[capabilitylist]]
          [;ACCESS=[(fileaccess)]]
          [;VS=volset:SPAN]
```

PARAMETERS

groupname — The name of the new group, composed of up to eight alphanumeric characters, beginning with a letter.

password — Group password, used for verifying log-on access only. Default is no password is assigned.

capabilitylist — A list of capability-class attributes, consisting of any or all of the following: IA, BA, PM, MR, DS, or PH.

This list imposes a limit on program files belonging to the group. A capability cannot be defined if it is not presently possessed by the group's account. However, if the account is altered later, the group could be left with excessive capabilities. Default is IA, BA.

filespace — Disc storage limit, in sectors, for the permanent files of the group. A group's *filespace* cannot be specified as greater than the corresponding limits currently defined for the group's account. However, an account's *filespace* can be changed later so that some of its groups are left with limits that exceed the new account limits. Default is same as account's *filespace*.

cpu — Limit on total cpu time, in seconds, for the group. This limit is checked only when a job/session is initiated--thus, the limit never causes a job/session to abort. The maximum value permitted is 2,147,483,647 seconds. A group's cpu limit cannot be specified as greater than the corresponding limit currently defined for the group's account. However, an account's cpu limit can be changed later so that some of its groups are left with a limit that exceeds the new account limits. Default is account's cpu limit.

connect — Limit on total session connect time, in minutes, allowed the group. This limit is checked only at log-on time. The maximum value allowed is 2,147,483,647 minutes. A group's connect limit cannot be specified as greater than the corresponding limit currently defined for the group's account. However, an account's connect time can be changed later so that

some of its groups can be left with a limit that exceeds the new account time. Default is the account's connect limit.

fileaccess — The restrictions on file access pertinent to this group. Default is R, X: ANY; A, W, L, S: AL, GU for Public Group (PUB); and R, A, W, L, X, S: GU for all other groups.

volset:SPAN — *Volset* is a volume set or class reference which, when fully qualified, is in the form *vcsid.groupname.acctname*, where *vcsid* refers to a previously-defined volume set or class definition.

SPAN indicates that *groupname* is to be inserted in the accounting directory of the specified volume set (*volset*). The specified volume set must have been previously mounted (via the MOUNT command) for the SPAN operation to succeed.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capability required:		AM

OPERATION

This command creates a new group within an account.

When a keyword parameter (such as ;PASS=) or keyword parameter group (such as ;PASS=password) is omitted from the :NEWGROUP command, the corresponding default value takes effect.

After the System Manager enters the :NEWACCT command, the new Account Manager can then redefine his own attributes, the Public Group, and new users and groups as desired. Note that the new Account Manager may only define or redefine groups and users within the set of parameters assigned at account creation time.

The Public Group is initially assigned a null password and the same capability-class attributes, permanent file space limit and cpu time limit as the account. Its initial security grants reading and program-execute access to all users who can pass the account's security, and appending, writing, locking, and saving access to account librarian and group users only.

:NEWGROUP

EXAMPLE

To create a new group named GROUP1, enter:

```
:NEWGROUP GROUP1
```

Default values are taken for all parameters.

TEXT DISCUSSION

Page 5-3.

:NEWUSER

Defines a new user.

SYNTAX

```
:NEWUSER username
      [;PASS = [password] ]
      [;CAP = [capabilitylist] ]
      [;MAXPRI = [subqueuename] ]
      [;LOCATTR = [localattribute] ]
      [;HOME = [homegroupname] ]
```

PARAMETERS

username The name of the user, composed of up to eight alphanumeric characters, beginning with a letter. (REQUIRED PARAMETER)

password User password. (This password is used for verifying log-on access only.) Default is no password is assigned.

capabilitylist List of capabilities, mutually separated by commas, permitted this account. Each capability is denoted by a two-letter mnemonic, as follows:

System Manager	=	SM
Account Manager	=	AM
Account Librarian	=	AL
Group Librarian	=	GL
Diagnostician	=	DI
System Supervisor	=	OP
Permanent Files	=	SF
Access of non-sharable I/O devices	=	ND
Use Volumes	=	UV
Create Volumes	=	CV
User Logging	=	LG
Process Handling	=	PH
Extra Data Segments	=	DS
Multiple RINS	=	MR
Privileged Mode	=	PM
Interactive Access	=	IA
Local Batch Access	=	BA

This parameter cannot specify capabilities that are not assigned to the user's account. However, if the account is changed later, the user could be left with capabilities that exceed those of the account; but, because the user's capabilities are always verified to be a subset of the account's at log-on time, the user is never granted a capability not possessed by the account. Default is SF, ND, IA, BA. Note that CV automatically gives the account UV capability.

subqueuename Name of the subqueue of highest priority that can be requested by any process of any job/session in the account, specified as "xS," where x is A, B, C, D, or E. The priority specified in :NEWUSER cannot be greater than that specified in :NEWACCT or :ALTACCT. If the user's account is changed later, however, the user may be left with excessive limits. The *subqueuename* defined for the user is checked against the *subqueuename* defined for his account at log-on time, and the lower priority of the two is used as the maximum priority restricting all processes of the job/session. Also, the priority requested by the user when he logs on is checked against the *subqueuename* defined for him, and he is granted the lower of these two values. Default is CS.

CAUTION

Processes capable of executing in the "AS" or "BS" priority classes can deadlock the system. System processes and other user processes can be completely locked out from execution by "AS" and "BS" priority processes. Care should be used in assigning these priorities.

localattribute The local attribute of the user, as defined at the installation site. This is a double-word bit map of arbitrary meaning which might be used to further classify users. While it is not involved in standard MPE security provisions, it is available to processes through the WHO intrinsic for use in the programmer's own security provisions. The bit map for the user local attributes must be a subset of the bit map for the account local attributes. The :NEWUSER command checks the user's local attributes with those of the account's. Default is double-word 0.

homegroupname The name of an existing group to be assigned as the homegroup for this user. If no homegroup is assigned, the user must always specify a group when logging on. Default is no homegroup is assigned.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required:		AM

OPERATION

This command defines a new user.

If the optional parameters or keyword parameter groups are omitted, the corresponding parameters specified in :NEWACCT are assigned by default.

EXAMPLE

To define a new user named LHSMITH, assign a password of SMITTY, and a homegroup of HOMEGPX, enter:

```
:NEWUSER LHSMITH;PASS=SMITTY;HOME=HOMEGPX
```

TEXT DISCUSSION

Page 5-4.

NEWUSER CREATION

1. LOG ON MGR.SS

2 NEWUSER_ BRUCE; HOME=PUB

3, FOR PASSWORD: NEWUSER_BRUCE; HOME=PUB;
PASS= ABC

change password → 4, ALTUSER BRUCE; PASS=(NEW PASSWORD)

5, DELETE PASSWORD=ALTUSER BRUCE; PASS=

:NEWVSET

Defines private volume sets and classes.

SYNTAX

```
:NEWVSET vsname
    ;MEMBERS = vname:type[,vname:type, . . .[vname:type]
    [;CLASS = vcname:vname[,vname,...[,vname]]
```

PARAMETERS

vsname Volume set name, consisting of from one to eight alphanumeric characters, beginning with a letter. This name is also given to the master volume of the set. MPE will implicitly reference *vsname* as *vsname.groupname.acctname*, where *groupname* and *acctname* are the log-on group and account. (REQUIRED PARAMETER)

vname:type Names of members of the volume set and the types of devices on which the members will reside. A maximum of eight volume names may be specified. The list of volume names must include the Master volume set name. The type must be specified with each volume member to define the type of disc required to accommodate the volume. Types of disc drives that may be specified are HP 7902, HP 7905, HP 7906, HP 7920, HP 7925, HP 7933, and HP 9895.

vcname:vname Volume class name, consisting of from one to eight alphanumeric characters, beginning with a letter. Each volume class consists of a volume class name and an associated list of volume names vname. The list of volume names must be a subset of the volumes comprising the volume set, and one of the volume names must be that of the volume set's master volume.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required:		CV

OPERATION

The :NEWVSET command creates private volume sets and classes. Once a volume set or volume class is defined, its definition is kept on disc in a volume set definition entry. A physical volume set is not created at the time the volume set is defined, only the members of the volume set and the corresponding storage types are defined to MPE.

EXAMPLE

To define a volume set with vsname USERSET; with members USERSET (master volume), MEM2, and type HP 7920; and class (vcname:vname) CLASS1:USERSET, MEM2, enter:

```
:NEWVSET USERSET;MEMBERS=USERSET:HP7920,MEM2:HP7920;CLASS=
CLASS1:USERSET,MEM2
```

TEXT DISCUSSION

Page 4-12.

:PURGEACCT

Removes an account and its groups (files) and users from the system's or the specified volume set's directory.

SYNTAX

:PURGEACCT *acctname* [;VS=*volset*]

PARAMETERS

acctname Name of the account to be deleted. (REQUIRED PARAMETER)

volset Volume set or volume class reference which, when fully qualified, is in the form

vcsid.groupname.acctname

where *vcsid* refers to a previously-defined volume set or volume class definition.

If *volset* is specified, the volume set or volume class must be mounted or the :PURGEACCT command will fail.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required:		SM

OPERATION

This command removes an account (and its groups (files) and users) from the system.

When the :PURGEACCT command is entered in a session, MPE displays a verification request. This aids in ensuring that the wrong account is not deleted accidently. The format of the verification request is:

ACCT *acctname* TO BE PURGED?

Response is YES or NO.

No verification message is printed when the :PURGEACCT is entered in a job.

The :PURGEACCT command removes every user not currently logged on and every group/file not in use. This could leave the account (containing residual elements only) still in the system. In such a case, if all users were purged, the account would exist but would be inaccessible for log on. To completely purge such an account, the :PURGEACCT command must be entered when the account is inactive. It is possible that the system would require a restart in order to close files which were left open.

The system account, SYS, cannot be purged. If such a command is entered, however, all groups except PUB will be purged. All users except the System Manager user are purged and all files which are not system files or not currently opened in the PUB group are purged.

EXAMPLE

To remove an account named ACCT1, the following dialog occurs:

```
:PURGEACCT ACCT1
ACCT ACCT1 TO BE PURGED? YES
```

TEXT DISCUSSION

Page 4-9.

:PURGEGROUP

Removes a group (and all files belonging to it) from the system's or the specified volume set's directory.

SYNTAX

:PURGEGROUP *groupname* [;VS=*volset*]

PARAMETERS

groupname Name of group to be removed. (REQUIRED PARAMETER)

volset Volume set or volume class reference which, when fully qualified, is in the form

vcsid.groupname.acctname

where *vcsid* refers to a previously-defined volume set or volume class definition.

If *volset* is specified, the volume set or volume class must be mounted or the :PURGEGROUP command will fail.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	Programmatically?	Yes
Breakable?		No
Capability Required:		AM

OPERATION

This command removes a group (and all its files) from the appropriate directory.

When this command is entered in a session, MPE prints the message:

GROUP *groupname* TO BE PURGED?

Response is YES or NO.

No verification message is printed if the :PURGEGROUP command is entered in a job. For this command to succeed, the group's home volume set must be mounted.

If no files in the group are in use, and the group itself is not in use, :PURGEGROUP removes the entire group. Otherwise, :PURGEGROUP removes every file not in use. In such a case, the :PURGEGROUP command must be entered again when neither the group nor any of its files are in use in order to completely purge the group. It is possible that the system would require a restart in order to close files which were left open.

The public group of the system account, PUB.SYS, cannot be purged. If the command is entered, however, all files which are not system files are purged.

EXAMPLE

To purge a group named GROUP1, the following dialog occurs:

```
:PURGEGROUP GROUP1
GROUP GROUP1 TO BE PURGED? YES
```

TEXT DISCUSSION

Page 5-7.

:PURGEUSER

Removes a user from an account.

SYNTAX

:PURGEUSER *username*

PARAMETERS

username Name of the user to be deleted. (REQUIRED PARAMETER)

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required:		AM

OPERATION

This command purges a user from the account.

When :PURGEUSER is entered in a session, MPE displays the following message:

USER *username* TO BE PURGED?

Response is YES or NO.

If entered from a job, the message is not displayed.

If the :PURGEUSER command references a user currently logged onto the system, that user is not purged and a message is output indicating this fact.

The original system manager, MANAGER.SYS, cannot be purged.

Purging a user may leave some files in an account without their creators. However, the Account Manager can access these files and purge them.

EXAMPLE

To remove a user named USER1, the following dialog occurs:

```
:PURGEUSER USER1
USER USER1 TO BE PURGED? YES
```

TEXT DISCUSSION

Page 5-7.

:PURGEVSET

Deletes an existing volume set.

SYNTAX

```
:PURGEVSET vsname
```

PARAMETERS

vsname Volume set name of an existing volume set, or volume class name of an existing volume class.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capability required:		CV

OPERATION

The :PURGEVSET command deletes a previously-defined volume set. A volume set will not be purged if the definition is currently in use (that is, a MOUNT, either explicit or implicit, has been issued using this definition). When :PURGEVSET is entered in a session, MPE displays the following message:

VOLUME SET/CLASS DEFINITION *vsname* TO BE PURGED?

EXAMPLE

To purge volume set USERSET, enter:

```
:PURGEVSET USERSET

VOLUME SET/CLASS DEFINITION USERSET TO BE PURGED (YES/NO) YES
```

TEXT DISCUSSION

Page 4-13.

:REPORT

Displays accounting information for log-on account and group.

SYNTAX

```
:REPORT [groupset] [,listfile]
[;VS = volset]
```

PARAMETERS

groupset Specifies the accounts and groups for which information is to be listed. The permissible entries and the capability required (shown in parentheses) are as follows: Account Manager is shown as AM; System Manager as SM.

groupdesig Reports on the specified group in the log-on account. Standard user can only specify his log-on group.

@ Reports on all groups in the log-on account (AM or SM).

groupdesig.acctdesig Reports on the specified group in the specified account (SM).

@.acctdesig Reports on all groups in the specified account (SM).

@.@ Reports on all groups in all accounts (SM).

Default: For standard user: his own group

For Account Manager: All groups in his own account.

For System Manager: All groups in all accounts.

NOTE

The characters @, #, and ? can be used as "wild card" characters in the groupset parameter. These wild card characters have the following meanings:

@ — specifies zero or more alphanumeric characters.

— specifies one numeric character.

? — specifies one alphanumeric character.

The characters can be used as follows:

n@ List all groups starting with the character *n*.

@*n* List all groups ending with the character *n*.

n@*x* List all groups starting with the character *n* and ending with the character *x*.

n##..# List all groups starting with the character *n* followed by up to seven digits.

?*n*@ List all groups whose second character is *n*.

n? List all two-character groups starting with the character *n*.

?*n* List all two-character groups ending with the character *n*.

listfile Actual file designator of output file to which information is to be written. Output may be re-directed with a :FILE back reference as follows:

```
:FILE LIST1;DEV=LP
:REPORT, *LIST1
```

Default is $STDLIST.

volset Directs the reporting of accounting information from the specified volume.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes. Aborts.
Capability Required:		AM or SM

OPERATION

Outputs total accounting information logged against groups and accounts. For the standard user, this information covers his own group only; an Account Manager may specify all groups in his account; and the System Manager may specify all groups in all accounts.

The information includes usage counts and limits for permanent file space (in sectors), cpu time (in seconds), and session connect time (in minutes). The file space usage count reflects file space used as of the present time, but the cpu time and connect time usage reflects these counts as they were immediately prior to the start of the current job.

The type of output written to *listfile* depends on the type of file (ASCII or binary) specified or implied. If *listfile* is an ASCII file, a standard ASCII listing is produced; on this listing, an unlimited quantity is denoted by a double asterisk (**). If *listfile* is a binary file (typically used to help in automatic processing of the report data), a 17-word record is written for each account/group.

On both ASCII and binary listfiles, the entry for each account is followed immediately by the entries for all of its groups.

EXAMPLE

To obtain accounting information for your group, enter the :REPORT command, as follows:

```
:REPORT
```

TEXT DISCUSSION

Page 4-6.

EXAMPLE

To reset the cpu use counter for all accounts in the system, enter:

```
:RESETACCT @,CPU
```

TEXT DISCUSSION

Page 4-8.

:RESETACCT

Resets to zero the running counts of CPU time or connect time accumulated by an account, and all groups within that account.

SYNTAX

```
:RESETACCT [   @     ][-,CPU     ]
           [acctname][,CONNECT]
```

PARAMETERS

@	Specifies that counters for all accounts (and all groups within the accounts) are to be reset. Default.
acctname	Specifies the name of a particular account (and all groups within the account), the totals of which are to be reset.
CPU	Specifies that only the CPU usage counter is to be reset. Default is both the CPU and connect time counters are reset.
CONNECT	Specifies that only the connect time usage counter is to be reset. Default is both the CPU and connect time usage counters are reset.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	YES
Breakable?		NO
Capability required:		SM

OPERATION

This command resets to zero the running counts of CPU or connect time accumulated by an account, and all groups within that account. Note that if all parameters are omitted, all counters (except File Space) for all accounts and their groups are reset.

:RESTORE

Returns the files (stored on magnetic tape or serial disc) to system.

SYNTAX

```
:RESTORE restorefile[;[fileset] [, . . . ,fileset] ]
    [;KEEP] [;DEV =device] [;SHOW] [;FILES = maxfiles]
    [;OLDDATE]
```

PARAMETERS

restorefile — Name of magnetic tape or serial disc file on which files to be restored exist. This file must be referenced in the back-reference format by using an asterisk. (This format references a previous :FILE command that has defined the file as a magnetic tape or serial disc file.)

A message is output on the system console requesting the Console Operator to mount the tape or serial disc medium identified by the restorefile parameter, and allocate the device. (REQUIRED PARAMETER)

fileset — Sets of parameters in any of the following formats:

filedesig.groupdesig.acctdesig	Restore the specified file in the specified group and account.
filedesig.groupdesig	Restore specified file in specified group.
filedesig	Restores the specified file.
@.groupdesig.acctdesig	Restore all files in specified group and account.
@.groupdesig	Restore all files in specified group.
@.@.acctdesig	Restore all files in all groups in specified account.
@	Restore all files in log-on group. Default for users.
@.@	Restore all files in all groups in log-on account. Default for Account Manager.
@.@.@	Restore all files in system. Default for System Manager.

NOTE

The characters @, #, and ? can be used as "wild card" characters in the fileset parameter. These wild card characters have the following meanings:

@	—	specifies zero or more alphanumeric characters.
#	—	specifies one numeric character.
?	—	specifies one alphanumeric character.

The characters can be used as follows:

n@	Restore all files starting with the character *n*.
@n	Restore all files ending with the character *n*.
n@x	Restore all files starting with the character *n* and ending with the character *x*.
n##. .#	Restore all files starting with the character *n* followed by up to seven digits (useful for restoring all EDIT/3000 temporary files).
?n@	Restore all files whose second character is *n*.
n?	Restore all two-character files starting with the character *n*.
?n	Restore all two-character files ending with the character *n*.

More than one fileset may be used with a single :RESTORE command. The number of filesets that may be specified with one :RESTORE command is limited as follows: up to 10 by account name; up to 15 by group name and account name; and up to 20 by file name, group name, and account name. The *fileset* is a positional parameter.

KEEP

Specifies that if a file referenced in the :RESTORE command currently exists on disc, the file on disc is kept and the corresponding tape or serial disc file is not copied into the system. Default: If an identically-named file exists in the system, that file is replaced with the one on tape or serial disc. If the identically-named file on disc is busy, however, the disc file is kept and the tape or serial disc file is not restored.

device

Specifies the device on which the file is to reside, entered in one of the following forms:

devclass
ldn

The device class (*devclass*) specifies the type of device. If *devclass* is specified, the file is allocated to any of the home volume set's volumes that fall within that device class.

The logical device number (*ldn*) specifies a specific device. If *ldn* is specified, the file will be allocated to that device only if one of the volumes in the home volume set currently occupies the device.

Default: MPE attempts to restore the file on a logical device compatible with the type/sub-type specified in the file's file label and the type/sub-type of the mounted home volume set. If this fails, an attempt is made to restore the file on any member of the home volume set. If this fails, the file is not restored.

SHOW — Request to list names of restored files. Default: Only total number of files restored, list of files not restored, (and the reason each was not restored), and the count of files not restored are listed. The listing is sent to $STDLIST (formal designator SYSLIST) unless a :FILE command is entered to send the listing to some other device. For example,

```
:FILE SYSLIST:DEV=LP
```

entered before the :RESTORE command would send the listing to a line printer.

maxfiles — Maximum number of files that may be restored. Default is 4000.

OLDDATE — Keeps the *accdate* and *moddate* on the files being restored unchanged. Default is to alter these dates.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes. Aborts.
Capability Required:		See OPERATION

OPERATION

Reads back into the system, on disc, a file or files stored offline by :STORE or :SYSDUMP commands. If you have System Manager or System Supervisor capability, you can restore any file from a :STORE tape or serial disc, assuming the account and group to which the file belongs and the user who created the file are defined in the system. If you have Account Manager capability, you can restore any file in your account (but cannot restore those with negative file codes unless you also have Privileged Mode capability). If you have standard user capability only, you can restore any file in your log-on account if you have SAVE access to the group in which the file belongs; you cannot, however, restore those files with negative file codes unless you have Privileged Mode capability. If the file to be restored is protected by a lockword, you must supply the lockword with the :RESTORE command unless you have System Manager, System Supervisor, or Account Manager (and you are restoring within your own account) capability. If you are logged on in an interactive session, MPE prompts you for omitted lockwords. However, if you are a System Manager, System Supervisor, or Account Manager (restoring within your own account), you are not required to provide lockwords. Any file belonging to a group whose home volume set has not been pre-mounted will not be restored.

The listing output by :RESTORE is sent to a file whose formal designator is SYSLIST. If not specified, this file is equated by default to the current list device ($STDLIST).

Before entering :RESTORE, you must identify *restorefile* as a magnetic tape or serial disc file with a :FILE command, as follows:

```
:FILE formaldesignator[= filereference] ;DEV = device [;REC = recsize]
     [;BUF = numbuffers]
     ;NOBUF
```

The *device* parameter must indicate the device class name or logical unit number of a magnetic tape or serial disc unit. The *recsize* parameter must be entered only if the record size of files being :RESTOREd differs from 4096 words. If recsize is greater than 4096, BUF = 1 or NOBUF must be specified. All other parameters for *restorefile* are supplied by the :RESTORE command. If you attempt to supply any of these parameters, MPE rejects the :RESTORE command.

EXAMPLE

To restore all files belonging to your log-on group from the restorefile named BACKUP, enter:

```
:FILE BACKUP;DEV=TAPE
:RESTORE *BACKUP;@;KEEP;SHOW
```

In response, the Console Operator receives a request to mount the tape identified as BACKUP. If a disc file satisfying the @ specification already exists in the system, it is not restored because the KEEP parameter was specified.

TEXT DISCUSSION

Page 6-121.

:RESUMELOG

Resumes logging following suspension caused by an error.

SYNTAX

```
:RESUMELOG
```

PARAMETERS

None

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required:		OP

OPERATION

This command resumes logging following suspension caused by an error.

When logging resumes, a special Log Record is displayed that denotes the number of log events (and corresponding records) missed, total records missed, total job initiation records missed, and total job session termination records missed.

EXAMPLES

Assume the system is on line and running with logging enabled. An error of a recoverable type occurs, resulting in the following error message to the system console:

```
ST/10:43/LOG FILE NUMBER 104 ERROR #46. LOGGING SUSPENDED
```

After the error is corrected, :RESUMELOG is entered:

```
:RESUMELOG
```

In response, the following messages are displayed on the system console:

```
ST/10:45/LOG FILE NUMBER 104. LOGGING RESUMED.
ST/10:45/LOG FILE NUMBER 104 ON.
```

TEXT DISCUSSION

Page 6-135.

:SETCATALOG

Causes the Command Interpreter to search a catalog of user-defined commands and to establish a directory entry for each command in the catalog, or to close an existing UDC file.

SYNTAX

:SETCATALOG[*catfilename*[,*catfilename*,...,[*catfilename*]]][;SHOW][;ACCOUNT / ;SYSTEM]

PARAMETERS

catfilename Catalog file name. Normally, this file would be prepared using the Editor. When more than one user-defined command resides in the catalog file, each command must be separated from its neighbors by one or more lines, the first character of which must be an asterisk.

SHOW Lists catalogs and UDC's as user-defined commands are initialized. This parameter is useful for listing additional information if there is an error in UDC initialization. SHOW lists each UDC as it is checked. If an error occurs, it is listed after the erroneous UDC.

ACCOUNT Specifies that the catalog file being defined or reset is at the account level. Requires AM capability.

SYSTEM Specifies that the catalog file being defined or reset is at the system level. Requires SM capability.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	NO
Breakable?		NO

OPERATION

The :SETCATALOG command causes the Command Interpreter to search a specially-prepared disc file for user-defined commands. The Command Interpreter then establishes a directory entry for each command in the file. When used without file names, this command may also be used to close an existing UDC file.

EXAMPLE

To create a user-defined command (UDC) and have the Command Interpreter establish a directory entry for it in the file MYCMNDS, enter:

```
:EDITOR

/ADD
      1   S              (UDC name)
      2   SHOWJOB        (MPE command)
      3   **             (Separates UDC's)
      4   R              (UDC name)
      5   RESUME         (MPE command)
      6   //
/KEEP MYCMNDS
/END

:SETCATALOG MYCMNDS
```

Thus, S and R are established as user-defined commands. If you enter S, the :SHOWJOB MPE command will execute; if you enter R, :RESUME will execute.

Example for altering catalog:

```
:SETCATALOG              (Closes existing UDC file)
:EDITOR

/TEXT MYCMNDS
/ADD
    6 **
    7 E
    8 EDITOR
    9//
/KEEP MYCMNDS
/END

:SETCATALOG MYCMNDS
```

TEXT DISCUSSION

Page 4-13 and 5-10.

:SHOWCATALOG

Lists user-defined command (UDC) files.

SYNTAX

:SHOWCATALOG [*listfile*]

PARAMETERS

listfile An arbitrary file name, causing the listing to be sent to a line printer. Default: If omitted, the listing is sent to $STDLIST. If specified, the listing is sent to device class LP (line printer) unless directed elsewhere with a prior :FILE command.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	YES
	Programmatically?	NO
Breakable?		YES (Aborts)

OPERATION

The :SHOWCATALOG command lists user-defined command (UDC) files and, specifies at which levels (USER, ACCOUNT, SYSTEM) the files have been defined.

EXAMPLE

To list UDC files and have the listing appear on the standard list device, enter:

```
:SHOWCATALOG
```

To have the listing sent to the line printer, enter:

```
:SHOWCATALOG LFILE
```

To have the listing sent to a disc device, enter:

```
:FILE LFILE;DEV=DISC
:SHOWCATALOG *LFILE
UDC CATALOG LIST SENT TO LIST FILE. (CI 1932)
```

TEXT DISCUSSION

Pages 4-15 and 5-10.

:SHOWLOG

Displays the current Log File's number and the amount (percentage) of available disc space currently used.

SYNTAX

```
:SHOWLOG
```

PARAMETERS

None

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required?		OP

OPERATION

This command displays the current Log File's number and the amount (percentage) of available file space currently used.

The information displayed appears in the following format:

LOGFILE LOG*xxxx* IS *yy*% FULL

where *xxxx* is the Log File number and *yy* is the percentage of file space used.

If the Logging System is disabled, MPE outputs the message:

NO LOGGING

If logging is enabled but currently suspended as a result of a managerial error, both messages appear.

EXAMPLE

To display the number and percent of file space used for the current Log File, enter:

:SHOWLOG

TEXT DISCUSSION

Page 6-135.

:SHOWQ

Displays information about the scheduling of processes and the contents of the various scheduling subqueues or the master queue.

SYNTAX

```
:SHOWQ
```

PARAMETERS

None

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes. Aborts.
Capability Required:		OP

OPERATION

This command displays information about the scheduling of processes and the contents of the various scheduling subqueues.

The information is divided into two major columns. In the right column, the processes are listed in high-to-low priority order. These processes currently (or will immediately) require the CPU in order to continue. The CPU will be given automatically to the highest priority process which is able to use it.

In the left column, the processes are listed in numerical order and are waiting on longer-term events. They will again contend for the CPU when those events occur.

After displaying the process queues, :SHOWQ prints the scheduling parameters currently in effect, and the minimum memory manager algorithm cycle value (*minclockcycle*).

On occasion, a process will appear in both columns. This indicates that the process was being moved to the right during the execution of :SHOWQ.

The following information is displayed for each process:

$$\begin{pmatrix} L \\ C \\ D \\ E \end{pmatrix} \left[\begin{pmatrix} M \\ U \end{pmatrix} \right] \textit{pin} \left[\# \begin{pmatrix} J \\ S \end{pmatrix} \textit{jobnum} \right]$$

where

L	—	is a linearly scheduled process on the AS, BS, or master queue.
C	—	is a circularly scheduled process on the CS queue.
D	—	is a circularly scheduled process on the DS queue.
E	—	is a circularly scheduled process on the ES queue.
M	—	is a job main process.
U	—	is a user process.
pin	—	is the process identification number for this process.
J	—	is the process executing in a batch job.
S	—	is the process executing from a session.
jobnum	—	is the job or session number.

EXAMPLE

To display the queues of processes within MPE, enter:

```
:SHOWQ
```

TEXT DISCUSSION

Page D-10.

:STORE

Stores disc files onto magnetic tape or serial disc in a manner which is recoverable by :RESTORE.

SYNTAX

```
:STORE { [fileset [-fileset]][, ...] } ;storefile
       { [!indirectfile]             }
       [;SHOW[=showoption[,showoption[,...]]]]
       [;FILES=maxfiles]
       [;DATE      {<=accdate}  ]
                   {>=moddate}
       [;ONERR[OR]={QUIT}       ]
                   {REDO}
```

PARAMETERS

fileset — Sets of parameters, entered in any of the following formats: (Default is @)

filedesig.groupdesig.acctdesig	Store specified file in specified group and account.
filedesig.groupdesig	Store specified file in specified group under log-on account.
filedesig	Store specified file under log-on group.
@.groupdesig.acctdesig	Store all files in specified group and account.
@.groupdesig	Store all files in specified group under log-on account.
@.@.acctdesig	Store all files in all groups in specified account.
@	Store all files in log-on group. Default.
@.@	Store all files in all groups under log-on account.
@.@.@	Store all files in system.

NOTE

The characters @, # and ? can be used as "wild card" characters in the fileset parameter. These wild card characters have the following meanings:

@ - specifies zero or more alphanumeric characters.
# - specifies one numeric character.
? - specifies one alphanumeric character.

The characters can be used as follows:

n@	Store all files starting with the character *n*.
@*n*	Store all files ending with the character *n*.
n@x	Store all files starting with the character *n* and ending with the character *x*.
n##..#	Store all files starting with the character *n* followed by up to seven digits (useful for storing all EDIT/3000 temporary files).
?*n*@	Store all files whose second character is *n*.
n?	Store all two-character files starting with the character *n*.
?*n*	Store all two-character files ending with the character *n*.

-fileset Name of a *fileset* or subfileset to be excluded from the store. (See EXAMPLE)

indirectfile An ASCII file which contains the desired parameters of the :STORE command. Each line of the *indirectfile* may contain both filesets and options, but only options may appear after the first semicolon on each line. An exclamation point (!) must precede this parameter.

storefile Name of destination tape or serial disc file onto which the stored files are to be written. Can be any magnetic tape or serial disc from the output set. This file must be named in a :FILE command and back referenced as shown in the following examples:

```
:FILE STORTAPE;DEV=TAPE

:STORE @.@;*STORTAPE
```

or,

```
:FILE STORDISC;DEV=SDISC

:STORE @.@;*STORDISC
```

SHOW Request to list files stored. For each file stored, the following information is printed on the user's terminal (default): filename, groupname, acctname, logical device number, file size (in sectors) and filecode. Note that as each file is stored, a message is printed on the user's terminal. Error messages are also displayed in the same way; thus, if a particular file is not stored due to its being "in use", disc error, etc., you may elect to abort the store process. (Formal designator is SYSLIST.)

showparm Any one or more of the following options (separated by commas):

SHORT	Same as default display above.
LONG	Causes SHOW to display record size, file type, EOF, file record limit, blocking factor, extents allocated and maximum extents of files stored.
DATES	Causes SHOW to display the creation date, last access date and last modification date of files stored.
SECURITY	Causes SHOW to display the creator and file access matrix of all files stored.
OFFLINE	Causes SHOW output to be sent to the system lineprinter as well as being displayed on the user's terminal. Note that the lineprinter output can be redirected to another output device by using a :FILE command (formal designator is OFFLINE). For example: :FILE OFFLINE;DEV=PP;CCTL

maxfiles	Maximum number of files that may be stored. If more than *maxfiles* are found, :STORE will print an error message and quit. Absolute maximum allowed is 32,767. Default is 4000.
<=*accdate*	Store only files not accessed since date specified.
>=*moddate*	Store only files which have been modified on or after date specified. When no date is specified all files specified by the *filesets* will be stored. The date is expressed mm/dd/yy.
ONERR[or]=QUIT	Causes "abnormal" termination of the :STORE program when a tape error is encountered.
ONERR[or]=REDO	Causes :STORE to close and rewind the current tape reel when a tape error is encountered. After rewind, a request for a new reel is generated on the user's terminal. Writing to the new reel begins at the same point in the *fileset* list as the previous (bad) reel. A message is sent to the operator's console advising of a tape error on the previously rewound reel.

DEN — Specifies the density that :STORE uses when writing to a multiple density tape drive. Default is 6250 for 7976 tape drives, and 1600 for all others.

NOTE

It is suggested that BUF always be used with tape, and NOBUF always be used with serial disc and command queueing tape. Default is BUF . (NOBUF must be used when REC = 8192 is specified.)

Other parameters for *storefile* are supplied by the :STORE command. If you attempt to supply them, MPE will reject the :STORE command. Note that :STORE can be invoked with the :RUN command, and the "INFO=" parameter can be used if desired, i.e.:

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

or,

```
:RUN STORE.PUB.SYS
```

If the BREAK key is pressed during a store operation, the operation continues while you interact with the command interpreter. Both ABORT and RESUME are useable within BREAK.

EXAMPLES

To copy all files in the group GP4X in your log-on account to a tape file named BACKUP, enter:

```
:FILE BACKUP;DEV=TAPE
:STORE @.GP4X;*BACKUP;SHOW
```

The console operator receives a request to mount the tape identified as BACKUP. A listing of the files stored appears on your standard list device as each is stored.

If a file is in use by the system, the file name is displayed, along with the message BUSY.

To STORE all files that have been modified since 8/30/82, with no error recovery:

```
:STORE @.@.@;*TAPE;DATE>=8/30/82;ONERROR=QUIT
```

To STORE all files that aren't in the PUB group and get the default show information:

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

To STORE files and get all information on both the line printer and $STDLIST:

```
:STORE @.@.@;*TAPE;SHOW=LONG,DATES,SECURITY,OFFLINE
```

To STORE using the specifications in the indirect file INDSTORE:

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

where the file INDSTORE might contain:

```
@.PUB.SYS,@.@.KSE;FILES=32767
@.@.SUPPORT;SHOW=LONG,DATES
```

TEXT DISCUSSION

Page 6-121.

USE

Available	In Session?	YES
	In Job?	YES
	In Break?	NO
	Programmatically?	YES
Breakable?		YES
Capability Required:		See OPERATION

OPERATION

Stores one or more disc files onto magnetic tape or serial disc. The :STORE command will store only those files whose home volume sets are mounted.

Users with System Manager or System Supervisor capability can store any user file in the system. Users with Account Manager capability can store any file in the account (but cannot dump those with negative file codes unless they have Privileged Mode capability also).

Before entering a :STORE command, you must identify *storefile* as a magnetic tape or serial disc file with a :FILE command written in the following format.

```
:FILE formaldesignator [=filereference];DEV=device
                       [;REC=recsize]
                       [;BUF[=numbuffers]]
                       [;NOBUF]
                       [;DEN=density]
```

where,

device	Must indicate the device class name or logical unit number of a magnetic tape or serial disc unit.
recsize	Sets the size of the blocks on the store tape. Record size may be any multiple of 256 words between 256 and 4096 words for tapes and up to 8192 for serial discs and command queueing tapes. Default is 4096 for non-programmatic calls to :STORE , and 1024 words for programmatic calls.
BUF	Causes the tape and disc I/O to overlap.
numbuffers	Number of buffers to be allocated for the file. This is an integer from 1 to 16. Default is 2.
NOBUF	Specifies that no buffers are allocated for the file.

:SWITCHLOG

Closes the current Log File, and creates and opens a new Log File.

SYNTAX

```
:SWITCHLOG
```

PARAMETERS

None

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required:		OP

OPERATION

This command closes the current Log File, and creates and opens a new Log File. The new Log File becomes the current Log File.

If logging is disabled when :SWITCHLOG is entered, no action is taken.

MPE displays the following information when :SWITCHLOG is entered:

LOGFILE LOG*xxxx* IS *yy*% FULL
LOG FILE NUMBER *zzzz* ON

where

xxxx — is the previous Log File number.

yy — is the percentage of file space used.

zzzz — current log file now opened (increment of 1 over last Log File number).

If logging is enabled but not currently suspended, MPE displays the message:

NO LOGGING

EXAMPLE

To switch logging to a new Log File, enter:

:SWITCHLOG

TEXT DISCUSSION

Page 6-135.

:SYSDUMP

Starts Configurator dialog and copies MPE to magnetic tape or serial disc.

SYNTAX

:SYSDUMP *dumpfile*[,*auxlistfile*]

PARAMETERS

dumpfile A back reference to a previous :FILE command that defines the file on which the modified or duplicate system is to be written. This back reference must indicate a magnetic tape file or a serial disc file. (REQUIRED PARAMETER)

auxlistfile Actual file designator of the output file (device) to which all listings requested during the Configurator/User dialog are written. The formal file designator used by the :SYSDUMP command executor for this file is SYSDLIST; the formal file designator used by the MPE Segmenter (when it is invoked to add or replace SL segments) for this file is SEGLIST. Default is $STDLIST.

NOTE

System-defined formal designators (such as SYSDLIST) cannot be used in the back reference format. To dump more than 16,000 files, set up a file equation :FILE SYSDGOOD;DISC=# *of records.*

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	No
	programmatically?	No
Breakable?		Yes. Suspends.
Capability Required:		OP

OPERATION

This command starts the Configurator/User dialog and copies MPE to magnetic tape or serial disc.

In addition to the names of the files to be used for *dumpfile* and *auxlistfile* output, you must specify what types of devices these files are to reside on. You do this by preceding the :SYSDUMP command with MPE :FILE commands defining each of the two files.

:SYSDUMP

The following format is used for the :FILE command:

:FILE *formal designator* [=*filereference*] ; DEV = *device*
[;REC=*recsize*] [;BUF=*numbuffers* | ;NOBUF]

device	Must indicate the device class name or logical unit number of a magnetic tape or serial disc unit.
recsize	Set the size of the blocks on the store tape. Record size may be any multiple of 256 words, between 256 and 4096 words for tape and up to 8192 for serial discs. Default is 4096 for non-programmatic calls, and 1024 words for programmatic calls.
BUF	Causes the tape and disc I/O to overlap.
NOBUF	Will not allow the tape and disc I/O to overlap. It is suggested that BUF always be used with tape, and NOBUF always be used with serial disc. Default is BUF.

Once the files are specified, you issue the :SYSDUMP command, preceding each filename with an asterisk (*) to indicate a back reference to the definitions in the :FILE commands.

EXAMPLE

To begin dialog with the MPE Configurator, the :SYSDUMP command is entered. At the end of the dialog, the MPE configuration requested is copied to the file named DUMP. Any listings requested during the dialog are output to the file named LIST. The first :FILE command specifies that DUMP is a magnetic tape file; the second :FILE command defines LIST as a line printer. (In the :FILE commands, the TAPE and LP parameters are device class names arbitrarily defined by a user during the previous configuration.)

```
:FILE DUMP;DEV=TAPE
:FILE LIST;DEV=LP
:SYSDUMP *DUMP,*LIST
```

TEXT DISCUSSION

Page 6-16.

Changes the filter and/or priority limits of the circular subqueues and/or the value of the memory manager algorithm threshold value.

SYNTAX

```
:TUNE [minclockcycle] [ { ;CQ                                                    } ]
                      [ { ;DQ = [base] [,[limit] [,[min][.[max]]]] . . .         } ]
                      [ { ;EQ                                                    } ]
```

PARAMETERS

minclockcycle Minimum value (in milliseconds) for replacement algorthm cycle through memory. If algorithm cycles through memory in less time than this value, memory allocation is delayed. (Thrash prevention mechanism.)

base Priority at which C, D, or E processes will begin in queue.

limit Worst (highest number) priority C, D, or E processes can attain.

min The minimum value indicating the length of time (filter value) in milliseconds which a process can have use of the CPU before its priority is reduced.

max The maximum value indicating the length of time (filter value) in milliseconds which a process can have use of the CPU before its priority is reduced. Note that the value of *max* must be greater than or equal to the value of *min*.

NOTE

The parameters *min* and *max* for processes executing the C queue refer to absolute limits of the "average short transaction" time (filter) which is recomputed after every transaction is completed. This average value is compared against the CPU time of each process to determine whether or not the process' priority should be decreased. For processes executing in the D or E queues, a fixed value is used rather than an average. The same value is used for both queues and is specified by setting *min* and *max* equal to the desired value. Both *min* and *max* must be specified, but if they are not given the same value, the value of *max* is used, (D queue and E queue only).

At least one of the parameters (*minclockcycle*, *base*, *limit*, *min*, or *max*), must be specified. The values for *base* and *limit* may range from 150 to 255. Parameters are positional, i.e., if *min* for a subqueue is changed without specifying *base* and *limit*, it must be preceded by two commas. If *minclockcycle* is omitted, the queue specification (CQ, DQ, or EQ) must be preceded by a semicolon.

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		No
Capability Required:		OP or SM

OPERATION

This command changes the filter or priority limits of a circular subqueue, and/or the memory manager algorithm threshold value, and is used primarily in on-line tuning of the system to best accommodate the current load.

> CAUTION! MISUSE OF THIS COMMAND CAN SIGNIFICANTLY DEGRADE SYSTEM OPERATING EFFICIENCY!

A CS process is given a priority of CBASE when it begins. (See Figure 2-1.) When a process stops (for disc I/O, terminal I/O, preemption, etc.), its new priority is determined so that it may be re-queued for the CPU. If the process has completed a transaction (a transaction is defined as the time between terminal reads), the priority becomes CBASE. The value of an "average short transaction" is then recalculated. If the CS process has not completed a transaction, and if the process has exceeded the average short transaction filter value since its priority was last reduced, the priority is decreased (but the value will not exceed the CLIMIT).

DS and ES processes begin at DBASE and EBASE respectively, and are rescheduled according to the same criteria as used for CS processes, with the exception that a fixed value (the value of *max* which has been specified for the subqueue) is used in place of the average short transaction value, which is used for CS processes only.

If the values specified for *max* in the CS, DS, and ES queues are too large, system response may become erratic. If they are too small, excessive memory management may result because process-swapping occurs too frequently. In either case, system performance is degraded. Recommended settings for *min* and *max* for the CS subqueue are 0 and 300, as it is desirable to favor short transactions in most cases. A value for *min* and *max* of 1,000 usually produces efficient system operation for the DS and ES subqueues.

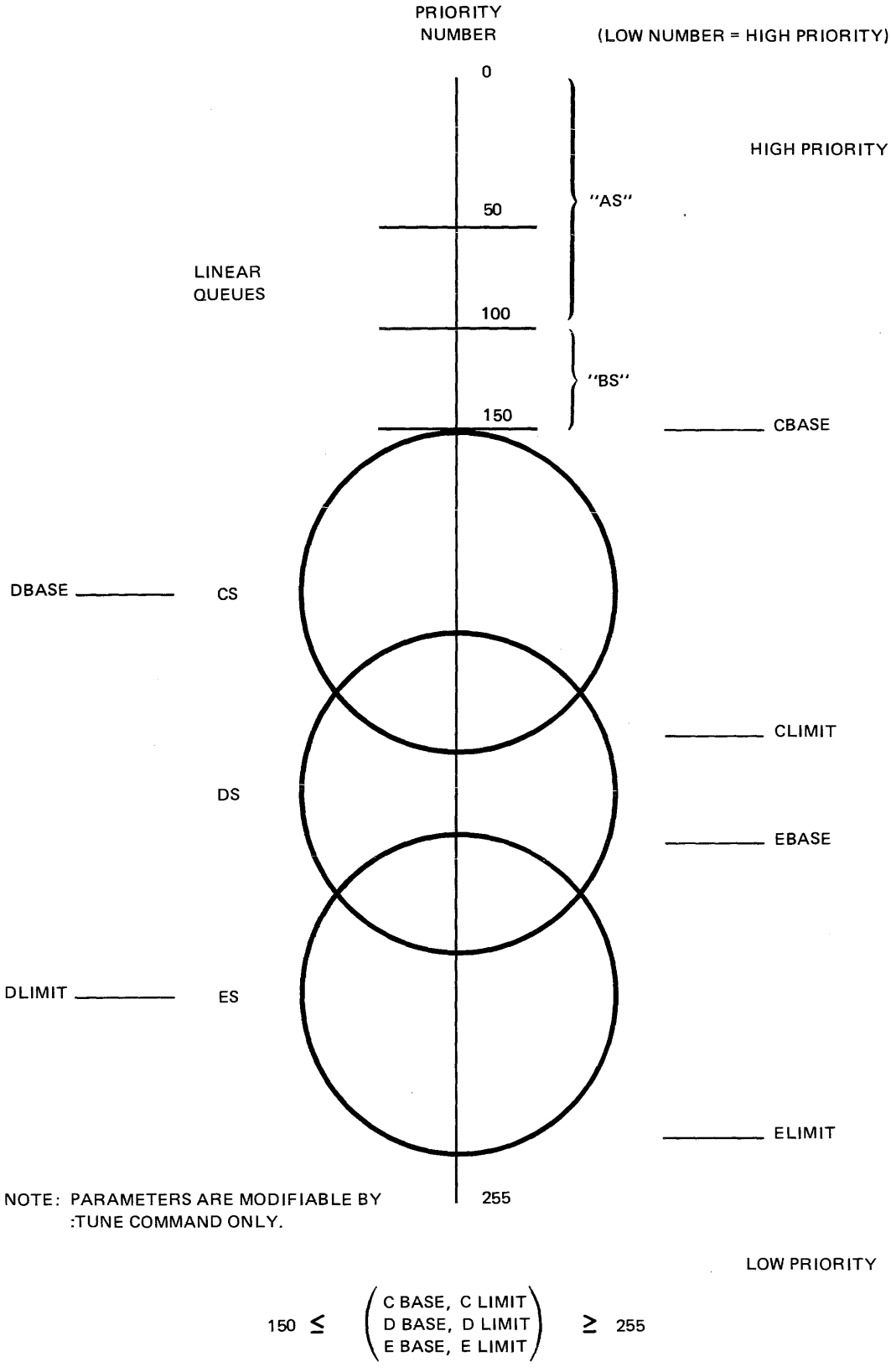


Figure 2-1. Master Queue Structure

The values assigned for limit, (CLIMIT, DLIMIT, etc.), specify the lowest priority (largest number) that the system can assign to a process in a particular queue.

The *minclockcycle* parameter is used by the memory manager to determine when thrashing (excessive memory management activity) is occurring. Making this value smaller increases the probability of thrashing. Recommended value for this parameter is 1000.

NOTE

The following default settings are in effect at system startup (COOLSTART, WARMSTART).

minclockcycle:	1,000		
CQ base:	152		
limit:	200		
min:	0	EQ base:	240
max:	300	limit:	253
		min:	1000
DQ base:	202	max:	1000
limit:	238		
min:	1000		
max:	1000		

EXAMPLE

To set the minimum clock cycle to 1,000, CQ base to 152, limit to 200, CQ minimum filter to 0, and CQ maximum filter to 300; and DQ base to 202, DQ limit to 238, DQ (and EQ) minimum (and maximum) filter to 1,000 enter:

```
:TUNE 1000;CQ=152,200,0,300 ;DQ= 202,238,1000,1000
```

TEXT DISCUSSION

None.

:VINIT

Accesses the Vinit subsystem to format and condition private volumes, serial discs and foreign discs.

SYNTAX

```
:VINIT [listdevice]
```

PARAMETERS

listdevice Name of the device to receive output generated by the Vinit subsystem. This parameter must back reference a :FILE command that names the device. Default is $STDLIST.

FUNCTIONS

> INIT vname, ldn [,vsname.groupname,acctname] [;GEN=genindex] Writes a PV label; uses last vs from INIT or PDEFN.

> FORMAT ldn Writes a scratch label; sets up FST, DFSM, DTT, DSCT, SPARES.

> SERIAL ldn Writes a serial disc type label.

> SCRATCH ldn [;RESET] Writes (resets) a scratch type label.

> COPY fromldn, toldn [;GEN=[genindex]] Copies volumes of the same type.

> DSTAT [ldn / ALL or @] Gives status of drives.

> PDEFN [* / vsname .group.account] List vs specified.

> PLABEL ldn Prints disc label.

> PFSPACE {ldn [;ADDR] / ALL} Prints disc free space map.

> PDTRACT ldn Prints defective track table contents, or defective sector table contents.

> COND ldn [;RECOVER] Condenses disc space holes; recovers lost disc space for PV

> DTRACK ldn Processes suspect tracks in defective track table.

> EXIT Exits the VINIT subsystem

> HELP Prints listing of VINIT commands with syntax.

:VINIT

USE

Available	In Session?	Yes
	In Job?	No
	In Break?	No
	Programmatically?	No
Breakable?		Yes (Suspends)
Capability Requires		SM or OP

OPERATION

The :VINIT command accesses the Vinit subsystem which contains several commands for the online conditioning and formatting of private volumes.

EXAMPLE

```
:FILE LIST;DEV=LP
:VINIT*LIST
```

TEXT DISCUSSION

Page I-26.

SYSTEM MANAGEMENT OVERVIEW

SECTION III

CAPABILITY SETS

The HP 3000 Computer System is used by a large variety of programmers, ranging from those who want to run simple applications programs in BASIC to system programmers who are actually modifying MPE. To protect the system and its users in general, users with System and Account Manager capabilities can limit access to special system capabilities only to those who fully understand their correct use. This is done through capability sets. Specifically, when a System Manager creates an account, he defines it for a capability set that determines whether or not users communicating with MPE through this account will be allowed certain functions. When an Account Manager defines the users of his account, he associates with each user some or all of the general account capabilities. Each capability set contains three types of attributes: user, file-access, and capability-class. A fourth attribute, the local attribute, may also be defined. The combination of these attributes determines the set of commands and intrinsics available to the user. This division of commands and intrinsics greatly simplifies use of the system from the standpoint of each user — it defines the extent to which the user must understand and interrelate with MPE, and permits a user to ignore aspects of MPE that do not apply to him.

Capability sets also are defined for groups by the Account Manager. Group capability sets contain only one type of attribute — the capability-class attributes. The capability set for a group may allow that group some or all of the capability-class attributes defined for the account to which the group belongs. The group capabilities relate to the user's capabilities as noted at the end of this section under PROGRAM CAPABILITY SETS.

As noted under PROGRAM CAPABILITY SETS, capability-class attributes also are associated with each program on a program file, and passed as parameters (in the command that prepares the program) to the MPE Segmenter.

USER ATTRIBUTES

The user attributes designate the general level at which the user interfaces with MPE. These attributes can be assigned in any combination, and define capabilities in addition to those of a standard user.

SYSTEM MANAGER ATTRIBUTE (SM). Grants the user the capability to manage the overall system and create the accounts within it. The first user with the System Manager Attribute is designated on the system tape furnished with the HP 3000 Computer System. The System Manager, in turn, can designate other users having the same or different capabilities.

The System Manager functions are to:

- Create new accounts.
- Modify accounts.
- Delete accounts.

If the program resides on a permanent file, the program's capability-class attributes should not exceed those defined for the *group* to which the program file belongs. If they do, the user will not be able to run the program when he attempts to do so.

Because the capability set is associated with the entire set of code segments being run (and hence with the process running them), all procedures, subprograms, and subroutines on those code segments have the same capability. For the same reason, a *user* need not have the same capabilities as the *programs* he runs.

The capability-class and program attributes are:

- Extra Data Segments (DS). This attribute allows users and programs to manage (create) extra data segments. Normally, a programmer uses these data segments for temporary storage of large amounts of data; thus his program global data area remains relatively small. The extra data segment is purged at the termination of the program. Extra data segments are managed through the GETDSEG, FREEDSEG, DMOVIN, DMOVOUT, and ALTDSEG intrinsics, which are discussed in the *MPE Intrinsics Reference Manual.*

- Privileged Mode (PM). The Privileged Mode attribute allows a user or program to access all MPE resources, including intrinsics, system tables, and privileged cpu instructions. Provided the programmer has this capability, he can run programs in a permanently privileged mode or a temporarily privileged mode through the GETPRIVMODE, GETUSERMODE, and SWITCHDB intrinsics. Also, the CREATE and GETPRIORITY intrinsics may be used. These intrinsics are discussed in the *MPE Intrinsics Reference Manual.*

  IMPORTANT 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.

- Multiple RIN (MR). This attribute allows the acquisition of more than one global RIN for a single process. Effectively, it allows exclusive use of more than one system resource simultaneously. You should be aware that resource deadlocking can occur if multiple resources are not managed correctly. This may seriously degrade system performance.

- Use the user logging facility (LG). This attribute permits the use of user logging commands.

SYSTEM MANAGEMENT OVERVIEW

SECTION III

CAPABILITY SETS

The HP 3000 Computer System is used by a large variety of programmers, ranging from those who want to run simple applications programs in BASIC to system programmers who are actually modifying MPE. To protect the system and its users in general, users with System and Account Manager capabilities can limit access to special system capabilities only to those who fully understand their correct use. This is done through capability sets. Specifically, when a System Manager creates an account, he defines it for a capability set that determines whether or not users communicating with MPE through this account will be allowed certain functions. When an Account Manager defines the users of his account, he associates with each user some or all of the general account capabilities. Each capability set contains three types of attributes: user, file-access, and capability-class. A fourth attribute, the local attribute, may also be defined. The combination of these attributes determines the set of commands and intrinsics available to the user. This division of commands and intrinsics greatly simplifies use of the system from the standpoint of each user — it defines the extent to which the user must understand and interrelate with MPE, and permits a user to ignore aspects of MPE that do not apply to him.

Capability sets also are defined for groups by the Account Manager. Group capability sets contain only one type of attribute — the capability-class attributes. The capability set for a group may allow that group some or all of the capability-class attributes defined for the account to which the group belongs. The group capabilities relate to the user's capabilities as noted at the end of this section under PROGRAM CAPABILITY SETS.

As noted under PROGRAM CAPABILITY SETS, capability-class attributes also are associated with each program on a program file, and passed as parameters (in the command that prepares the program) to the MPE Segmenter.

USER ATTRIBUTES

The user attributes designate the general level at which the user interfaces with MPE. These attributes can be assigned in any combination, and define capabilities in addition to those of a standard user.

SYSTEM MANAGER ATTRIBUTE (SM). Grants the user the capability to manage the overall system and create the accounts within it. The first user with the System Manager Attribute is designated on the system tape furnished with the HP 3000 Computer System. The System Manager, in turn, can designate other users having the same or different capabilities.

The System Manager functions are to:

- Create new accounts.
- Modify accounts.
- Delete accounts.

- List accounts, groups, and users for record purposes.
- List file attributes.
- Obtain account reports.
- Store and restore any or all files on the system.
- Create private disc volume sets.
- Alter private disc volume sets.
- Delete private disc volume sets.
- Define user-defined commands (UDCs) for use by all system users.

ACCOUNT MANAGER ATTRIBUTE (AM). Allows the user to manage all users and groups within the account. The first manager for each account is designated by the System Manager when the account is created. The Account Manager, in turn, can assign this attribute to other users in the account.

Within the account, the Account Manager functions are to:

- Create new groups and users.
- Modify groups and users.
- Delete groups and users.
- List groups and users for record purposes.
- Obtain account reports.
- List account files.
- Store and restore any or all account files..
- Create private disc volume sets (must have Create Volumes (CV) capability).
- Alter private disc volume sets (must have CV capability).
- Delete private disc volume sets (must have CV capability).
- Define user-defined commands (UDCs) for use by all account users.

SYSTEM SUPERVISOR ATTRIBUTE (OP). Allows the user to have day-to-day external control of the system. It allows the user to manage scheduling subqueues, alter the system configuration, maintain the system and user logging facilities, and display various items of system information. The System Supervisor attribute can be assigned by the System Manager.

The System Supervisor functions are to:

- Manage the system log file facility.
- Exercise scheduling control over processes.
- Permanently allocate/deallocate code in virtual memory.
- Obtain certain system reports and information.
- Back up the operating system.
- Modify the operating system parameters.
- Save any or all files for archival purposes on magnetic tape or serial disc.

ACCOUNT LIBRARIAN ATTRIBUTE (AL). Can be assigned to grant a user special file-access modes for maintenance of files within his account. For example, an Account Librarian Attribute may be used to designate users who can purge (but not create or alter) files within the account. (File-access modes such as read-access or write-access, are discussed in Section IV.) This attribute is assigned by users with the Account Manager Attribute.

- Process Handling (PH). This attribute allows the direct creation of other processes by the running user process. In addition, it allows process suspension, inter-process communication, and process deletion. The programmer can use any of the following intrinsics (discussed in the *MPE Intrinsics Reference Manual*):

CREATE	SUSPEND	RECEIVEMAIL
TERMINATE	GETORIGIN	GETPRIORITY
KILL	MAIL	FATHER
ACTIVATE	SENDMAIL	GETPROCINFO
		GETPROCID

- Communications Software (CS). This attribute allows users to obtain exclusive access to a communications device such as a DSN/RJE line or a DSN/DS line. It is required for use of the DSN/RJE subsystem.

All the above class and program capabilities are optionally assignable. For the following two capabilities, one or both must be assigned to all users (except console operators).

- Interactive Access (IA). Allows access to MPE in an interactive (session) mode.

- Batch Access (BA). Allows access to MPE in a batch processing (job) mode.

Most users have only interactive and batch (IA, BA) standard capabilities. Either the interactive or the batch access attribute is required to communicate with MPE. The optional capabilities (DS, PM, MR, PH) are independent and can be assigned in any combination. You are reminded, however, that the more powerful the optional capability, the more hazardous its misuse is to the system integrity.

The operating system makes a check for some or all of the capabilities described above at various times during a job or session. These checks are made at:

- Log-on time. The system checks the user capability against that of the account. The user logging on cannot exceed the capability of the account.

- Program preparation time. The Segmenter checks the user capability against those capabilities requested in the preparation command. The capabilities of the program being prepared cannot exceed the capability of the user who is requesting the preparation.

- Program run time. The capabilities of the program being run cannot exceed the capabilities of the group in which the program resides. In addition, certain intrinsics require extra capability. When a program is run and calls an intrinsic requiring extra capability, a check is made against the program to ensure it has the extra capability required.

LOCAL ATTRIBUTES

Local attributes are contained in a double-word (i.e., two 16-bit computer words) bit map, used only for special applications that require further unique classification of users by other users. Each of the 32 bits represents one local attribute, so 32 local attributes are available. Local attributes are used with APL/3000 and checked by the :ALTUSER and :NEWUSER commands with the account assigned attributes. The local attributes are defined by System or Account Managers and are used by accounts or groups for any purpose desired. You may access the local attributes in your program via the WHO intrinsic described in the *MPE Intrinsics Reference Manual*.

Account and user local attributes are represented by bit maps, and are compared on a bit-by-bit basis. It is necessary to assign a bit pattern of user local attributes which is a subset of the account local attributes' bit pattern. For example, suppose the System Manager assigns account local attributes with a value of 12. The bit pattern for 12 is 1100, so a user may have local attributes which are represented by 8 (bit pattern 1000) or 4 (bit pattern 0100) or 0 (bit pattern 0000) or 12 (bit pattern 1100). If an attribute bit is set for a user, and the corresponding attribute bit is not set for the account, then the user local attribute is not valid; so, in the example above, a user may not specify local attributes represented by 6, because the bit pattern for 6 (0110) is not a subset of 12 (1100).

ACCOUNT/GROUP/USER ORGANIZATION

When you log on to MPE, two basic elements must be defined: an identifiable unit to which system resources (such as disc file space and central processor time) are allocated and charged, and a local set (domain) of disc files accessible by you. The basic unit to which resources are assigned is the *account*; this is the major "billable unit" in MPE. Associated with each account is a unique file domain, a set of *users* who can access MPE through this account, and a set of *groups* which partitions the account's accumulated resources and divides its file domain into private sub-domains.

Each account is defined, modified, and deleted by commands issued by a user with the System Manager capability. Every account is identified by a name. Optionally, a password can be associated with the account to validate a user's ability to access MPE under this account at log-on time. A maximum priority also is associated with the account; this designates the highest priority at which any process within this account can be scheduled. Limits are assigned for maximum disc file space, central processor time, and on-line connect time permitted each account; running counts of the use of these resources are maintained for billing purposes. To maintain an account, the user acting as System Manager grants a user the Account Manager capability. This account-managing user may in turn assign the same capability (or some subset of these capabilities) to other users in his account.

The users and groups associated with each account are defined by commands issued by the account-managing user. Each user is identified by a name (unique to this account) and optional password. He is assigned a maximum allowable priority for his processes, which cannot exceed the maximum account priority. Each account possesses a public group (called PUB), to which all of its users have read and program-execution access, in addition to other groups that may be covered by various security provisions. Each group is identified by a name unique within its account. and optionally, by a password used to validate access to the group and its files at log-on time. As with an account, limits are assigned for the maximum disc file space, central processor time, and on-line connect time usable by a group; and running counts of resources used by the group are maintained. File space is always charged to the group containing the file, rather than the group to which the user who created the file was logged on.

Any MPE installation can contain several accounts; each account can have several users and groups associated with it; each group can possess several files which constitute a subset of the file domain. When the user logs on, he specifies the account, user and group names (and, if, required, the account, user, and group passwords). Furthermore, any file in a group may also be protected by a *lockword* required at any time the user accesses the file during the course of his job or session (in addition to standard file security mechanisms described later).

Each user can be associated with a *home group* by the user managing his account. If the user does not specify a group when he logs on, he is given the home group by default. If the Account Manager does not assign a user a home group, that user must always specify a group when he logs on.

Once the standard user has established communication with MPE, if the normal (default) system security provisions are in force, the user has unlimited access to all files in his log-on group and home group. Furthermore, he can read, and execute programs residing in, files in the public group (PUB) of his account or in the public group of the system account (PUB.SYS). He cannot, however, access other files in the system in any way.

The normal MPE security provisions can be overridden at the account, group, or file level, (by System Manager, Account Manager, or standard users, respectively) to provide more or less restriction to users. Users with special capabilities are generally subject to fewer restrictions.

A user with the MPE System Manager Capability can create, modify, and delete accounts. With each account, the System Manager User also creates a user with the Account Manager Capability and a Public Group (named PUB). The Account Manager User can then create, modify, and delete groups and users within his account. (The Public Group is a group whose file library is normally accessible for reading and program execution to all users within the account.)

TYPICAL ACCOUNTS

To illustrate the various ways in which accounts, groups, and users can be defined and organized at an HP 3000 Computer System site, several examples are presented. Notice that these are only examples, and are not meant to imply formal organizational rules.

A MULTI-DIVISION COMPANY

Suppose that a certain multi-divisioned company has a corporate HP 3000 Computer System. The company has many separate operating divisions but we will use the General Systems Division (GSD) to show the details of the accounting structure. (See figure 3-1.)

The System Manager user assigns the name GSD to an account and an Account Manager user name MGR. The Account Manager then logs onto the system and defines the groups for the accounting department (ACCTG), marketing department (MKTG), and engineering department (ENGR). He also assigns users to the various groups as detailed in the figure. In this case, the structure within the account follows the functional departmental scheme.

IN-HOUSE SERVICE BUREAU

An in-house service bureau, providing both batch and interactive capabilities, serves different users within the same company. The manager of the Data Center, having the System Manager Capability, creates an account for each department (Engineering, Marketing, Accounting, and other departments) in the company. For each account, an Account-Manager User assigns as users all programmers in his department, and assigns them all standard capabilities plus the following optional capabilities: Processing-Handling (PH), Data-Segment handling (DS), and Multiple RINS (MR). The Account Manager user also defines the groups for the account.

As an example of a typical account, the account for the Engineering Department defines all programmers in that department as users. This account contains the following groups:

- Public Group, containing utility programs, for which all users of the account have execute or read access only. (Types of access are discussed later in this section.)
- Data-Collection Group, for which all users have read, write (append only), save, and file-locking access.
- Common Group, used for miscellaneous department overhead. Unrestricted access; no passwords necessary.

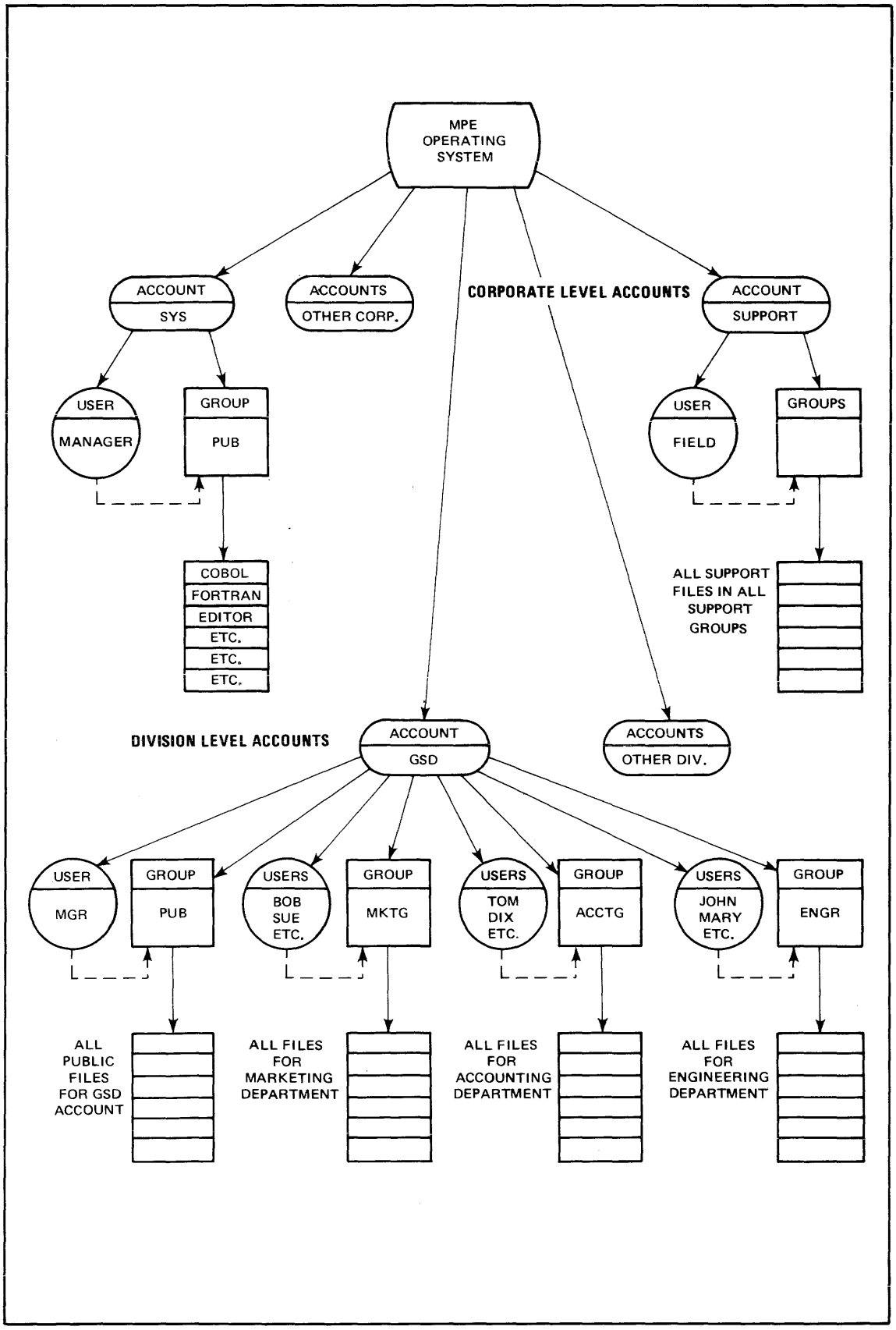


Figure 3-1. Accounting Structure in a Corporation

- Project Groups (one per project). Access restricted to project members.

- Private Groups (one for each programmer with private password).

The MPE Accounting System would be used, to some extent, for internal bookkeeping. This system monitors permanent file space and central-processor time used by job/sessions for accounts and groups, and interactive terminal connect-time used by sessions for accounts and groups.

COMMERCIAL SERVICE BUREAU

A commercial service bureau, also providing batch and interactive capabilities, offers its services to various outside companies (customers). An account is established for each customer. Users are given standard capabilities but no optional capabilities. The group organization is similar to that for the in-house service bureau described above. The MPE Accounting System is used more fully in the commercial service bureau, however.

SCIENTIFIC RESEARCH CENTER

In a research center where scientific programmers are using the system for batch and interactive applications, the following accounts are available:

- System account, used by all users.

- Data-processing account, containing groups allocated as required, used by all research users. For this account, all users have the standard capabilities and the following optional capabilities; Process-Handling, Data-Segment Handling, and Multiple RIN's.

- Special accounts, used by agencies such as the Personnel and Accounting Offices. Access is restricted to members of these agencies.

In this system, accounting is not critical, and the MPE Accounting System could be used minimally.

ACCOUNTING INFORMATION

The MPE Accounting System keeps track of various system resources used by each account and group; these resources are total permanent file space and central processor time (accumulated by jobs and sessions), and terminal connect time (accumulated by sessions). Furthermore, limits can be set for the maximum use of these resources at the account level (by System Manager users issuing the :NEWACCT or :ALTACCT commands) and at the group level (by Account Manager users issuing the :NEWGROUP and :ALTGROUP commands).

When a job/session is in progress, MPE maintains counts of the time-resources used by that job/session. As the job/session is logged off, its total time-resource use counts are used to update the time resource-use counters for its log-on account and group. When another job/session attempts to log-on to the same account (and perhaps group), and the central-processor time limit or session connect-time limit has been exceeded by the previous job/session at the account (or group) level, access is refused.

When a request is made to save a file, or add an extent to an existing file, and this action would result in exceeding the permanent file space limit at either the account or group level, the request is denied. (File space is always charged to the group containing the file, rather than the group where the user who created the file was logged-on.)

FILE SECURITY

Associated with each account, group, and individual file, is a set of security provisions that specifies any restrictions on access to the files in that account or group, or to that particular file. (Note that these provisions apply to disc files only.) These restrictions are based on three factors:

1. Modes of access (reading, writing, or saving, for example).

2. Types of users (users with Account Librarian or Group Librarian capability, or creating users, for example) to whom the access modes specified are permitted.

3. Use private volumes. Allows users to access files residing on private disc volume sets.

The security provisions for any file describe *what modes of access* are permitted to *which users* of that file.

The access modes possible, the mnemonic codes used to reference them in MPE commands relating to file security, and the complete meanings of these modes are listed in table 3-1.

The types of users recognized by the MPE security system, the mnemonic codes used to reference them, and their complete definitions are listed in table 3-2.

Users with System or Account Manager capability bypass the standard security mechanism. A System Manager has unlimited file access to any file in the system (R, A, W, L, X: ANY), but can save files only in his account (S: AC); an Account Manager user has unlimited access to any file within the account (R, A, W, L, X, S: AC). One exception is that in order to access a file with a negative file code, the Account Manager must also have the Privileged Mode (PM) capability.

The user-type categories that a user satisfies depend on the file he is trying to access. For example, a user accessing a file that is not in his home group is not considered a group librarian for this access even if he has the Group Librarian user attribute.

Notice that in order to extend a file, either W or A access to that file is required.

NOTE

In addition to the above restrictions in force at the account, group, and file level, a file lockword can be specified for each file. Users then must specify the lockword as part of the filename to access this file.

Table 3-1. File Access Modes

ACCESS MODE	MNEMONIC CODE	MEANING
Reading	R	Allows users to read files.
Locking	L	Permits a user to prevent concurrent access to a file by himself and another user. Specifically, it permits use of the FLOCK and FUNLOCK intrinsics, and the exclusive-access option of the FOPEN intrinsic, all described in *MPE Intrinsics Reference Manual*.
Appending	A	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.
Writing	W	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 previously.
Saving	S	Allows users to declare files *within a group* permanent, and to rename such files. This ability includes the creation of a new permanent file with the :BUILD command.
Executing	X	Allows users to run programs stored on files, with the :RUN command or CREATE intrinsic.

The security provisions for the account and group levels are managed only by users with the System Manager and the Account Manager capabilities respectively, and can only be changed by those individuals. The provisions are assigned and altered by the *fileaccess* parameter in the :NEWACCT, :ALTACCT, :NEWGROUP, and :ALTGROUP commands.

ACCOUNT-LEVEL SECURITY

The security provisions that broadly apply to all files within an account are set by a System Manager user when he creates the account. The initial provisions can be changed at any time, but only by a System Manager user.

Table 3-2. User Type Definitions

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

At the account level, five access modes are recognized:

Reading (R)
Appending (A)
Writing (W)
Locking (L)
Executing (X)

Also, at the account level, two user types are recognized:

Any User (ANY)
Account Member (AC)

If no security provisions are explicitly specified for the account, the following provisions are assigned by default:

- For the system account (named SYS), through which the System Manager user initially accesses the system, reading and executing access are permitted to all users; appending, writing, and locking access are limited to account members. (Symbolically, these provisions are expressed as follows:

  (R,X:ANY; A,W,L:AC)

In this format, colons are interpreted to mean ": . . . is permitted only to . . .", or " . . . is limited to . . .". Commas are used to separate access modes or user types for each other. Semicolons are used to separate entire access mode/user type groups from each other.)

- For all other accounts, the reading, appending, writing, locking, and executing access are limited to account members. (R,A,W,L,X:AC).

GROUP-LEVEL SECURITY

The security provisions that apply to all files within a group are initially set by an Account Manager user when he creates the group. They can be equal to or more restrictive than the provisions specified at the account level. (The group's security provisions also can be less restrictive than those of the account — but this effectively results in *equating* the group restrictions with the account restrictions, since a user failing security checking at the account level is denied access at that point, and is not checked at the group level.) The initial group provisions can be changed at any time, but only by an account-managing user for that group's account.

At the group level, six access modes are recognized:

Reading (R)
Appending (A)
Writing (W)
Locking (L)
Executing (X)
Saving (S)

Also, at the group level, five user types are recognized:

Any User (ANY)
Account Librarian User (AL)
Group Librarian User (GL)
Group User (GU)
Account Member (AC)

If no security provisions are explicitly specified, the following provisions apply by default.

- For a public group (named PUB), whose files are normally accessible in some way to all users within the account, reading and executing access are permitted to all users; appending, writing, saving, and locking access are limited to account librarian users and group users (including group librarian users). (R,X:ANY; A,W,L,S:AL,GU).

- For all other groups in the account, reading appending, writing saving, locking, and executing access are limited to group users. (R,A,W,L,X,S:GU).

FILE-LEVEL SECURITY

When a file is created, the security provisions that apply to it are the default provisions assigned by MPE at the file level, coupled with the user-specified or default provisions assigned to the account and group to which the file belongs. At any time, however, the creator of the file (and *only* this individual) can change the file-level security provisions, as described in the *MPE Commands Reference Manual* (see :ALTSEC command). Thus, the total security provisions for any file depend upon specifications made at all three levels — the account, group, and file levels. A user must pass tests at all three levels — account, group, and file security, in that order — to successfully access a file in the requested mode.

The creating user may also temporarily :RELEASE a file and then :SECURE it again through the use of commands described in the *MPE Commands Reference Manual.*

If no security provisions are explicitly specified by the user, the following provisions are assigned at the file level by default:

- For all files, reading, appending, writing, locking, and executing access are permitted to all users. (R,A,W,L,X:ANY).

Because the total security for a file always depends on security at all three levels, a file not explicitly protected from a certain access mode at the file level may benefit from the default protection at the group level. For example, the default provisions at the file level allow the file to be read by any user — but the default provisions at the group level allow access only to group users. Thus, the file can be read only by a group user.

In summary, the default security provisions at the account, group, and file levels combine to result in *overall* default security provisions as listed in table 3-3. Stated another way, when the default security provisions are in force at all levels, the standard user (without any other user attributes has:

- Unlimited access (in all modes) to all files in his log-on group and home group.
- Reading and executing access (only) to all files in the public group of his account and the public group of the System Account.

The important file security rules may be defined as follows:

- Users can create files in their own account.
- Only the creator can modify a file's security.
- If a lockword is present on a file, then it is required in order to access the file.
- Account Managers have unlimited access to the files within their account.
- System Managers have unlimited access to any file, but can save files only in their account.
- :RELEASE allows unlimited file access.
- :RELEASE does not change the file security settings; it only bypasses them temporarily.

Table 3-3. Default Security Provisions

FILEREFERENCE	FILE	ACCESS PERMITTED	SAVE ACCESS TO GROUP
filename.PUB.SYS	Any file in Public Group of System Account.	(R,X:ANY; W:AL,GU)	AL,GU
filename.groupname.SYS	Any file in any group in System Account.	(R,W,X:GU)	GU
filename.PUB.*accountname*	Any file in Public Group of any account.	(R,X:AC; W:AL,GU)	AL,GU
filename.groupname.accountname	Any file in any group in any account.	(R,W,X:GU)	GU

SYSTEM MANAGER CAPABILITIES

SECTION IV

THE SYS ACCOUNT

When Hewlett-Packard delivers an HP 3000 Computer System, there are two accounts on the system. These accounts are SYS and SUPPORT.

The SYS account is present on the system because system files such as PROGEN, INITIAL, SYSDUMP, the system Segmented Library (SL.PUB.SYS), and supported subsystems, such as COBOL and FORTRAN, must belong to *some* account. The SYS account has been assigned to contain these programs and code segments. (Subsystem programs and files are not part of the operating system, but are in the SYS account for convenience.)

The SUPPORT account is primarily used by Hewlett-Packard and your local Customer Engineering support group to distribute and maintain system software and all supported software utilities and verification programs. Since software support is not considered a user function, the SUPPORT account is not discussed in this manual. Your Customer Engineer can provide any information you may require concerning this special account.

Thus the primary initial account for the user is the SYS account. It is always present initially, and is expected to be present thereafter at all times.

Since every account has a *group* to contain files and a *user* to access those files, the SYS account also has an initial group and user. The group is named PUB; the user is named MANAGER.

The PUB group contains all system files, subsystem files, and Segmented Library procedures.

The System Manager user has full control over the SYS account, and in addition, has the ability to create all other accounts and managers that will be required for the installation. He has this ability because of the *System Manager optional capability*. It is this System Manager optional capability that will be discussed in detail in this section.

THE SYSTEM MANAGER

The System Manager has the most extensive capabilities of any user on the system. By definition, he has all the capabilities of the standard user as well as those that are unique to System Manager. In addition, he usually (but not necessarily) has Account Manager capability for the SYS account. (Account Manager capabilities are discussed in Section V.)

Those functions that are unique to the System Manager capability are:

- Creation of new accounts, and Account Managers.
- Modification of account capabilities and status.
- Deletion of entire accounts with associated groups, users, and files.
- Listing of any or all files on the system.

- Obtaining account reports of system usage and resources.
- Storing/restoring on magnetic tape or serial disc any or all files on the system.
- Defining private disc volume sets and classes.
- Altering private disc volume sets and classes.
- Deleting private disc volume sets and classes.
- Defining user-defined commands (UDCs) known to all system users.
- Creating the device-user association table.
- Creating an OPERATOR .SYS user

The System Manager normally logs on in session mode with:

```
:HELLO MANAGER.SYS
```

Initially there is no password on either the account, the group, or the user. Once logged on, the manager user can define new accounts and account manager users in order to set up the various billable units. He also can alter the SYS account, PUB group, and MANAGER user. He may create a new account with any or all of the attributes of the SYS account. If the System Manager capability (SM) is assigned to another manager user of a different account, then that manager user will also be a System Manager. It is thus possible that several different "system" accounts and several "system managers" could exist simultaneously on one system. However, it is recommended that only *passwords* be added or altered. This is especially true during the first few days of system use, since the SYS account is specially known to and by the operating system; if alterations are made to the SYS account before a working knowledge of the system is acquired, it could result in loss of system integrity.

In any case, the SYS account must always reside on the system. Normally (as when delivered by Hewlett-Packard), the SYS account is assigned the maximum attributes. If the SYS account is altered for any reason, it must have the following attributes as a minimum:

capabilitylist — It is recommended that all capabilities be assigned to the SYS account.

passwords — Account, group, and user *passwords* may be assigned by the System Manager to the SYS account as desired.

filespace — Must be unlimited for the SYS account. Additionally, the system is periodically changed by Hewlett-Packard; in most cases, more disc space is required in the SYS account when changes are made.

cpu — Central processing time is not counted against system files and subsystems which are run on behalf of users in other accounts. *Cpu* process time in the SYS account is mainly kept track of by customers to determine the accounting of the System Manager and System Supervisor. If there are other users and groups in the SYS account, it may be useful to keep track of the *cpu* resource used by those users and groups, and to impose a limit. If this limit is exceeded, System Manager users will be warned when logging on; other users will be denied access.

connect Like the *cpu* processing attribute, a *connect* limit may be useful for accounting purposes, particularly if other groups and users are in the SYS account. If this limit is exceeded, a System Manager user will be warned when logging on; other users will be denied access.

fileaccess It is not advisable to alter the file accessing parameters of the SYS account and PUB group. Normally, all users in the system can read and execute any file in the PUB group of the SYS account. Since all systems programs and all supported subsystems reside in the PUB group of the SYS account, it is reasonable to allow all users the READ and EXECUTE access. If a System Manager desires to place user programs and files in the PUB group of the SYS account and requires limited access to those programs and files, it is better to place lockwords on those programs and files rather than to alter the access attribute.

NOTE

The system programs residing in the PUB group of the SYS account are normally executed by the operating system on behalf of users. If users should attempt to :RUN such system programs as PROGEN or PFAIL, for example, the operating system will reject the attempt.

subqueuename The default subqueue assigned to the SYS account is CS. The CS subqueue is the highest priority subqueue under which all users should access the system; this includes System Managers. Allowing any user to execute on a higher priority will interfere with normal system processing and could cause degradation of system performance.

localattribute The default assigned to the SYS account is a double-word 0. The System Manager can assign the SYS account any local attribute he desires.

In order to perform his job as the System Manager user, the manager user has the following commands for his exclusive use:

:ALTACCT	
:LISTACCT	(For all accounts)
:NEWACCT	
:PURGEACCT	
:REPORT	(For all accounts)
:RESETACCT	

These commands, which are the domain of the System Manager user, will be discussed in detail in the remaining part of this section. Continuing related examples will be used to illustrate typical situations.

DEFINING ACCOUNTS

To create a new account and an associated Account Manager user and PUB group, the System Manager logs onto the system and enters the :NEWACCT command and any optional parameters.

Suppose that Bill Williams of the accounting department wants an account for his personnel. He desires 25000 sectors of disc space so that some of his programs can create large permanent files. He will be the Account Manager and desires an account password of "USONLY". He further states that he wants unlimited connect and cpu processing time, and desires to access magnetic tapes and the line printer. As the System Manager, you determine with Bill that his account will be named "ACCTG" and the Account Manager name will be "BILL". You then log onto the system as MANAGER.SYS (with passwords, if any) and assign the new account as follows:

:NEWACCT ACCTG, BILL; FILES=25000

To ensure that the account, PUB group, and Account Manager user are correct, you enter the following:

:LISTACCT ACCG
(A listing of the account will be printed; see Appendix D.)

:LISTGROUP PUB.ACCTG
(A listing of the PUB group of ACCTG will be printed; see Appendix D.)

:LISTUSER BILL.ACCTG
(A listing of the Manager user will be printed; see Appendix D.)

After decoding the account/group/user printed information, you would have the following information about the new account:

Account

acctname	ACCTG
password	Null (none)
filespace	25000 sectors available
cpu	Unlimited time
connect	Unlimited time
capabilitylist	AM, AL, GL, SF, ND, IA, BA
fileaccess	(R, A, W, L, X: AC)
subqueuename	CS
localattribute	Null (double-word 0)

Group

groupname	PUB
password	Null (none)
filespace	25000 sectors available
cpu	Unlimited time
connect	Unlimited time
capabilitylist	IA, BA
fileaccess	(R, X: ANY; A, W, L, S: AL, GU)

User	
username	BILL
password	Null (none)
subqueuename	CS
localattribute	Null (double-word 0)
homegroup	PUB
capabilitylist	AM, AL, GL, SF, ND, IA, BA

After all the decoding is completed, you become aware that the account password was inadvertently omitted. You correct the problem by entering:

:ALTACCT ACCTG; PASS=USONLY

All the other attributes described for the account ACCTG have remained the same except that the account *password* is now "USONLY". After the establishment of the account, it is a good idea to discuss the Account Manager responsibilities with Bill Williams before he and any of his personnel begin to use the account. (See Section V for Account Manager discussion.)

After several weeks, Bill Williams returns to you and informs you that he must have the capability for his account to obtain extra data segments. You then log onto the system and enter:

:ALTACCT ACCTG; CAP=AM, AL, GL, SF, ND, IA, BA, DS

You must enter the complete list of CAP= *capabilities* in order to retain the existing capabilities. All other attributes of the account remain the same. In addition, all groups and all users within the account have remained the same. That is, the PUB group and all other groups as well as the Manager user BILL and all other users' attributes have *not* been changed to include the extra data segment (DS) capability. It is now the responsibility of the Account Manager user to alter any groups and users (including himself) to include the extra data segment (DS) capability for those who need it.

DISPLAYING ACCOUNTING INFORMATION

A System Manager can list the attributes (previously specified in the :NEWACCT or :ALTACCT commands) for one or all accounts in the system by using the :LISTACCT command as in the previous examples. Also, by using the :LISTGROUP and :LISTUSER commands as in the previous examples, he can list the attributes for one or all groups and users in the system.

Each listing appears as an octal dump showing one or more entries. Each entry is headed by A= (for accounts), G= (for groups), or U= (for users), followed by an appropriate name and three or four lines of octal coded information. Examples are given in Appendix D of this manual.

There is also a supported utility program, LISTDIR2, which will format all of the octal information that is printed by the commands. Privileged information within LISTDIR2 is limited to System Managers and Account Managers.

In addition to the list commands, the System Manager also may use the :REPORT command to obtain account and group reports. As an example, suppose that your site is a timesharing business which bills the various customers monthly. You bill each customer by their account and the following resources used:

- Permanent disc space being used.
- Cpu processing time.
- Connect time in minutes since last billing cycle.

Today is the end of the billing period and you as System Manager must obtain a list of resources used by each account (customer). In addition, there is a delinquent account, DIAG, which you intend to purge from the system in order to deny access by that customer; you therefore want an individual report for that account. In order to obtain these reports of account usage, the System Manager will use the :REPORT command.

When obtaining information for a private volumes group, two :REPORT commands must be entered to get the complete accounting information. Two commands are necessary because when a group is bound to a private volume set the accounting information for that group is kept in two places. The CPU time counts and limits and the connect counts and limits are kept in the system directory, while the filespace counts and limits are kept in the directory on the group's home volume set.

Therefore, to obtain the accounting information from the system directory the following should be entered:

```
:REPORT groupset
```

This gives the CPU and connect time and limits for all groups but only the filespace counts and limits for those groups which are not bound to a private volume set. Note that the total filespace count for an account obtained from this command does not include the filespace counts for private volume groups.

To obtain these counts the following command must be used:

```
:REPORT groupset;VS=volset
```

This command lists only the group and accounts which are in the directory for the private volume set specified. The only valid information obtained in this report is the filespace counts and limits.

The :REPORT command for the System Manager is an expanded version of the command that is available to the standard user for displaying total accounting information for a user's log-on account and group; this is discussed in the *MPE Commands Reference Manual.*

The type of output written to *listfile* depends on the type of file (ASCII or binary) specified or implied. If *listfile* is an ASCII file, a standard ASCII listing is produced; on this listing, an unlimited quantity is denoted by a double asterisk (**). If *listfile* is a binary file, (typically used to help in automatic processing of the report data), a 17-word record is written for each account/group. This record is decoded as shown in table 4-1.

Table 4-1. Report Record Format

WORD	CONTENTS
0	Type of entry, where: 1 = A group entry. 2 = An account entry.
1-4	Account or group name, left-justified and padded with blanks.
5-6	Permanent file space count (in sectors).
7-8	Permanent file space limit (in sectors).
9-10	Central processor time count (in seconds).
11-12	Central processor time limit (in seconds).
13-14	Connect time count (in minutes).
15-16	Connect time limit (in minutes).

On a binary *listfile*, counts and limits are double-word integers; an unlimited quantity is denoted as %17777777777.

As an example, the System Manager logs onto the system in the usual way, entering passwords as appropriate. Since this is to be a very long printed report, the manager first enters a file command to direct output to the high speed line printer. Assuming that device 6 is the line printer, the command is:

```
:FILE LINEPR; DEV=6
```

The System Manager then enters the command to obtain the accounting report as follows:

```
:REPORT @.@, *LINEPR
```

The report for the entire accounting system is then printed on the high speed line printer. After the report is done, the System Manager then desires to have a report on the delinquent account DIAG printed on his session device. He enters:

```
:REPORT @.DIAG
```

The report, covering all groups in the account DIAG, is printed on the ASCII session device.

RESETTING ACCOUNTS

After the foregoing reports are obtained and the printed accounting report is delivered to the billing department, the System Manager desires to reset the cpu processing time and connect time counts for the new billing period. He enters the following command:

```
:RESETACCT @
```

This command without any optional parameters will reset the cpu processing time and the connect time for all accounts and groups on the system. The end result is that all cpu and connect counts will be set to zero. Users on the system will now begin accumulating cpu and connect time for the new billing period. Note that this command does not change the allowable limits for each account and group — only the counters for actual use.

STORING FILES

Continuing with the preceding example, the System Manager now desires to purge the DIAG account. Before he does this, however, he must ensure that any users, groups, and files are saved. He does this in anticipation of the customer paying his delinquent account, which would require reinstatement of the account. The System Manager saves the account files by using the :STORE command.

The System Manager user, like the System Supervisor user, has the ability to dump and retrieve any or all files in the system by using the :STORE and :RESTORE commands. These commands are defined in Section II. In this example, the System Manager first enters the file command to direct the output to a magnetic tape unit as follows:

```
:FILE SAVDIAG; DEV=TAPE
```

(Assume that device class TAPE is associated with a magnetic tape device.) The System Manager then mounts a magnetic tape on the magnetic tape device and enters the following command:

```
:STORE @.@.DIAG; *SAVDIAG; SHOW
```

(At some installations, the console operator would mount the tape after the :STORE command is entered.) After the console operator intervention is made to allocate the proper magnetic tape device, all files in all groups of the DIAG account will be stored on the magnetic tape. In addition, the optional parameter SHOW was included by the System Manager so that a record of all files for the account can be kept. Once the entire file structure has been saved, the System Manager is now ready to delete the account.

PURGING ACCOUNTS

Now the System Manager can delete the DIAG account and still be able to restore it should the customer become reinstated on the system. He enters the command as follows:

```
:PURGEACCT DIAG
```

The system will then request verification from the user as follows:

```
ACCOUNT DIAG TO BE PURGED? (YES/NO)
```

The System Manager responds with a YES answer. At this time, all files, all users, all groups, and all user and account level UDCs within the DIAG account, and the DIAG account itself, are removed from the system.

JOB/SESSION PRIORITY REQUEST

A special capability granted to System Managers and System Supervisors is the use of the optional parameter HIPRI in the :HELLO and :JOB commands. (See the *MPE Commands Reference Manual*.) This is a request for maximum job-selection priority, and will cause the job or session to be started regardless of current job fences set by the operator.

LISTING FILES

The System Manager as well as the ordinary user has the ability to list files. The System Manager can, however, list any or all files on the system. The listing produced can include passwords for those files, depending on the option that is used by the System Manager. The command is the :LISTF command, with parameters for use by the System Manager. See the *MPE Commands Reference Manual* for a discussion of this command.

PRIVATE VOLUMES

The Private Volumes Facility allows users who have the Use Volumes (UV) capability to access removable disc volumes. Users access a file by specifying a fully-qualified designator which causes the system to access, and mount if necessary, the appropriate set of volumes on which the file resides. The System Manager, and Account Managers who possess the Create Volumes (CV) capability, can define, alter, and delete private volume sets and classes.

The following paragraphs describe the Private Volumes Facility.

DISC DOMAINS

Under private volumes, the disc drives configured into the system are divided into two classes, or domains:

- System domain devices.
- Non-system domain devices.

Disc drives are determined to be system or non-system domain devices according to the contents of the Volume Table. Specifically, the system assumes that each volume added to the Volume Table resides on a system domain drive. Each time the system is cold loaded, the Initiator checks that all volumes in the Volume Table are physically present on the system. (The volumes in the table may reside on any suitable drive except for the system disc which must always be unit 0 on logical device number 1 (ldn 1); this volume cannot be moved.) Once the Initiator determines that all volumes in the Volume Table are present, all other disc drives that are configured into the system are considered non-system domain devices for private volume, serial disc, and foreign disc use. Thus, briefly, system domain devices cannot be used for mounting private volumes and non-system domain devices can.

SYSTEM DOMAIN DEVICES. System domain devices are considered by MPE to be permanently mounted. The domain can consist of drives which do not have removable packs and/or those with removable packs which the System Manager does not wish to have used as private volume devices. The system disc always is mounted on a system domain device.

NON-SYSTEM DOMAIN DEVICES. Non-system domain devices always have removable surfaces, are allocated by the System Manager for private volumes, serial disc, or foreign disc use, and can be mounted and dismounted during normal system operation. Private volumes, serial discs and foreign discs always are mounted on non-system domain drives.

NOTE

There are special considerations which apply to HP 7905 and 7906 discs. For these devices, the fixed platter must be configured in the system domain or not configured at all. The removable platter may be configured in the system domain or as a private volume. However, if the fixed platter is in the system domain and the removable platter is a private volume, changing the private volume will require switching the entire disc drive off-line. This will prevent access to the system platter until the drive is returned to the

on-line state. This is especially important if the system domain platter is configured as logical device number one. In this case the system domain platter contains the system directory and virtual memory and turning the disc drive off will stop all swapping and file input/output. For these reasons, it is recommended that both platters of a multi-platter device be configured in the system domain. This is especially true if the device is logical device number one.

VOLUME SET

A volume set is a set of not more than eight removable disc volumes which share a common file directory.

All files of a volume set are identified in the volume set's file directory. They are unique to that volume set, and cannot be shared among other volume sets. Those volumes permanently mounted on system domain devices comprise the System Volume Set and are named SYSVS by MPE.

Private volume sets and classes are allocated at the group level within an account. Each group is assigned to only one volume set — its home volume set; however, a volume set, if so defined, may be shared by more than one group within an account, or even among several accounts.

Each volume set has its own accounting directory. The system directory is the accounting directory for the system volume set. For user, or non-system volume sets, the appropriate account and group entries are included in its directory and in the system directory.

MASTER VOLUME. The master volume is that volume of a volume set which contains the volume set's accounting directory. It also contains a table of volumes for the volume set.

VOLUME CLASS. A volume class is a subset of the volumes in a volume set and must include the master volume of the set. It is accessed as a unit and is identified by a unique volume class name. It is the smallest volume unit that can be referenced by a user. A single volume can be referenced only if it is the master volume for the set or class. When a user references a volume class, therefore, he is notifying MPE that only a portion of a defined volume set need be mounted to satisfy his anticipated file-access requests. If a volume class consists of more than one volume, all volumes must be mounted in order to access the volume class.

HOME VOLUME SET. A home volume set is the volume set assigned to a group when the group is created with the :NEWGROUP command or altered with the :ALTGROUP command. Files belonging to the group are located on the home volume set. The home volume set is the set implicitly referenced by a user logging on under the group. The home volume set for the group, however, need not be mounted until such time as a user attempts to access file space in the set. At this time, MPE generates a console request for the Console Operator to mount the set. A user can explicitly request that the volume set be mounted before any file access is attempted, and can release the set after he is through with it. See the MPE Commands Reference Manual for a description of user mount and dismount requests.

All temporary files will be built on the home volume set. This requires that the volume set be physically mounted.

VOLUME SET/CLASS DEFINITION

Volume sets and classes are defined (created) by System Managers who have the create volumes (CV) capability and by Account Managers who also have the create volumes (CV) capability. Each volume set, and volume class within the set, is distinguished by a unique name assigned with the vsname (volume set name) and vcname (volume class name) parameters of the :NEWVSET command. Once defined, a volume set or volume class can be referenced by any user with the Use Volumes (UV) capability.

A volume set/class definition is of the form:

vsname.groupname.acctname

or

vcname.groupname.acctname

where

vsname — is the volume set name, consisting of from one to eight alphanumeric characters, beginning with a letter.

vcname — is the volume class name, consisting of from one to eight alphanumeric characters, beginning with a letter.

groupname.acctname — are the names of the group and account to which the volume set/class is to be assigned.

To define a volume set having the following characteristics

group	GRP1
account	ACCT1
volume set name	USERSET1
members	MEM1, MEM2, USERSET1
type	HP 7920
classes	CLASS1:MEM1,USERSET1

you would log on with

username.ACCT1,GRP1 (Must have CV capability)

and enter:

```
:NEWVSET USERSET1;MEMBERS=USERSET1:HP7920,MEM1:HP7920,&
: MEM2:HP7920;CLASS=CLASS1:USERSET1,MEM1
```

ALTERING A VOLUME SET DEFINITION

To alter home volume set USERSET1 in order to add a new class, CLASS3, with a new member MEM4, you would log on with

username.ACCT1.GRP1

and enter:

```
:ALTVSET USERSET1;ADDCLASS=CLASS3:USERSET1,MEM4;EXPANDSET=&
: MEM4:HP7920
```

DELETING A VOLUME SET DEFINITION

To delete the volume set USERSET1, you would enter:

```
:PURGEVSET USERSET1
```

MPE will request verification that USERSET1 is to be purged by displaying the following message:

VOLUME SET/CLASS DEFINITION USERSET1 TO BE PURGED (YES/NO)

The volume set will not be purged if any member is in use when the :PURGEVSET command is entered.

DEFINING UDC COMMANDS

A user-defined command or UDC is a procedure built from one or more standard MPE commands. When you combine several MPE commands into a single procedure and assign a name to that procedure, you create a UDC. In this way it is possible to enter the UDC name and cause several MPE commands to be executed. UDCs defined by the system manager can be made known to and available to all system users.

User-defined commands may also be defined by the Account Manager for use in the account, and by individual users for private use. When identically named UDCs occur at different levels, the user level UDC takes precedence over the account level UDC which, in turn, takes precedence over system level UDCs. When identically named UDCs occur at the same level, the UDCs take precedence in the order in which they are specified in the :SETCATALOG command.

NOTE

If a UDC references one of several identically-named UDCs, the order of precedence may not be the same as specified above. A good practice is to give your UDCs distinct names.

UDC SYNTAX

The format of a user-defined command is fully described in Section III of the *MPE Commands Reference Manual.*

The Editor can be used to modify commands stored in a file, if that file is not being accessed. Note, however, that a UDC file, once entered into a catalog with the :SETCATALOG command, cannot be modified and kept under the same name with the Editor. You must first release the file from the UDC catalog by issuing the :SETCATALOG command with no parameters (which releases all files in the UDC catalog).

ESTABLISHING A UDC DIRECTORY

The :SETCATALOG command is used to inform MPE that a specified file name contains user defined commands. The Command Interpreter than searches the file and establishes a directory for each command contained in the file. The file name is stored in a directory (COMMAND.PUB.SYS) of all UDC users. A user must have lock access specified for the group in which the file is located to use the :SETCATALOG command; a user must also have read access to the file. Note that the default file access for the PUB group gives lock access only to users with AL, GL, or home group (GU) capability.

The UDC directory, COMMAND.PUB.SYS, must exist for the :SETCATALOG command to properly execute. If this file does not exist on your system, you must build it with a record size of 20:

```
:BUILD COMMAND.PUB.SYS;REC=20,6,F,BINARY
```

If you want to have the file named SMUDC1 searched by the Command Interpreter for all users in the system, and have the user-defined commands contained therein entered into a catalog, enter:

```
:SETCATALOG SMUDC1;SYSTEM
```

Note that once a UDC file is entered into a catalog with the :SETCATALOG command, that file cannot be purged with the :PURGE command nor modified and kept under the same name with the Editor. Any attempt to do either of these two operations will result in an error message (Exclusive Violation).

In order to purge or modify such a file, it must be removed from the catalog by issuing the :SETCATALOG command with no file names, as in:

```
:SETCATALOG;SYSTEM
```

The effect is that all UDC files are removed from the catalog (the files are not purged, merely deleted from the catalog). This will take effect only after all users (except yourself) that had been accessing the file have logged off or issued a :SETCATALOG themselves. To re-enter UDC files into the catalog, re-enter the :SETCATALOG command for these files.

LISTING UDCs

To list the contents of a UDC file catalog, enter:

:SHOWCATALOG

MPE responds by providing a list of UDCs by assigned name and level (USER, ACCOUNT, or SYSTEM).

MANAGING EXPANDED CONSOLE CAPABILITIES

Console operator functions are sometimes associated with one specific user — the console operator. The use of expanded console capabilities increases system console accessibility and enables the console operator to disperse console functions to other users who may require them. The fact that console commands are processed by the MPE command interpreter enables the console operator to assign individual users the ability to execute certain console commands. The system console itself can be moved to any terminal on the system by use of the :CONSOLE command. Users with SM (system manager) capability may use the :CONSOLE command even if they have not been allowed the command, enabling them to retrieve the console at any time. Users can be associated with specific devices thereby enabling them to use the commands that control each device. Users can also be granted the ability to use job control commands on their own jobs.

Although not directly responsible for assigning these capabilities to users, the System Manager plays an important role in managing expanded console capabilities. Specifically, the System Manager must:

- create a device-user association table, and
- create a special user called OPERATOR.SYS.

CREATING A DEVICE-USER ASSOCIATION TABLE

The device-user association table identifies the devices which may be assigned by the system manager to other system users. The table is generated by running the utility ASOCTABL.PUB.SYS which creates a file called ASOCIATE.PUB.SYS, the device-user association table. This utility is described in the MPE System Utilities Manual.

In response to prompts issued by the utility, the System Manager enters association statements in the form:

deviceclassname = user.acct [,...,*user.acct*]

The variable deviceclassname is the appropriate device identifier assigned to the device to be associated. User is the name of the user to be allowed to associate with the device. Acct is the name of the user's account. More than one user may be allowed to associate with a device in the table. @.Acct may be used to indicate all users within an account; @.@ may be used to indicate all users on the system.

For example, to allow device class "LPX" to be associated by a user named ALDEN in account SALES, enter:

LPX = ALDEN.SALES

Once the table is created, any of the users listed can associate with the special deviceclass. Although more than one user may have *permission* to associate with a deviceclass in the table, a user can actually associate to a deviceclass only if none of the devices in that deviceclass are associated to anyone else.

CREATING OPERATOR.SYS

The creation of the user OPERATOR.SYS should be done at the time the system is first brought up. Creation of this user is a one-time activity, as OPERATOR.SYS remains on the system for use thereafter.

The first step in creating OPERATOR.SYS is to determine the home group and system capabilities that will be needed by the user. The home group used can be PUB or another group, even one created particularly for OPERATOR.SYS. The capabilities you decide to assign should take into account the kinds of activities that the user of OPERATOR.SYS will need to perform. It is recommended that the following capabilities be assigned as a minimum: OP to allow ;HIPRI logons and IA allowing the user to log on and run programs. Additional capabilities that will usually be required are: UV for handling private volumes, SF for storing files, ND for using tapes, and BA for batch jobs. Additional capabilities can be assigned if they are needed, however, care should be taken not to jeopardize the integrity of the system.

To create OPERATOR.SYS, log on with:

```
:HELLO MANAGER.SYS
```

then enter the following information, substituting the home group and capabilities you've selected for assignment:

```
NEWUSER OPERATOR.SYS;HOME=PUB;CAP=OP,IA,UV,BA,SF,ND,LG
```

ACCOUNT MANAGER CAPABILITIES

SECTION V

When an account is created by a System Manager, it always has an account name, a public group (PUB), and the first user of the account. By definition, the first user of a newly-created account is the Account Manager, and thus will have the Account Manager (AM) capability attribute. This attribute allows this user certain expanded capabilities over other users within the account. The Account Manager has less capability than the System Manager but does have final control over the overall use of the account. The Account Manager is automatically assigned a homegroup of PUB.

Suppose that a System Manager has created an account for Tom Smith, having used the command

```
:NEWACCT SMITH,TOM;CAP=AM,AL,GL,CV,SF,ND, UV,IA,BA;LOCATTR=184
```

The account structure would be as follows:

ACCOUNT	
acctname	SMITH
password	Null (none)
filespace	Unlimited
cpu	Unlimited time
connect	Unlimited time
capabilitylist	AM, AL, GL, CV, SF, ND, UV, IA, BA
fileaccess	(R, A, W, L, X: AC)
subqueuename	CS
localattribute	184

GROUP	
groupname	PUB
password	Null (none)
filespace	Same as account files
cpu	Same as account time
connect	Same as account time
capabilitylist	UV, IA, BA
fileaccess	(R, X: ANY; A, W, L, S: AL, GU)

USER	
username	TOM
password	Null (none)
subqueuename	Same as account
localattribute	Same as account
homegroup	PUB
capabilitylist	Same as account

It is important for an Account Manager (like TOM in the present example) to remember that any changes required to the account must be made by the System Manager. The Account Manager, however, can perform the following functions within his own account:

- Create new groups and users.
- Modify the attributes of groups and users.
- Delete entire groups, and files within the group.

- Delete users.
- Obtain reports on resources used within the account.
- List any or all groups, users, and files within the account.
- Store and restore any or all files within the account.
- Define private disc volume sets and classes for groups within the account. (Must have CV capability.)
- Alter private disc volume sets and classes for groups within the account. (Must have CV capability.)
- Purge private disc volume sets and classes for groups within the account. (Must have CV capabililty.)
- Define user-defined commands (UDCs) which are known to all users in the account.

After the SMITH account has been created, the Account Manager can log onto the system with

```
:HELLO TOM .SMITH
```

As can be seen, there is no password on this account (the System Manager did not assign one), or the group, or the user. The Account Manager can define new groups and users in order to divide the account logically. The major limitation to the number of groups and users that can be defined is the physical size of the System Directory. The System Directory resides on the system disc and its size, in sectors, is determined when the system is configured for a given installation by the System Supervisor. See Appendix E for a discussion of disc space utilization. The directory contains the structures for all accounts, groups, users, and files. It is therefore possible to have insufficient disc space allocated for the directory if a great many accounts, groups, users, and files will be required. The Account Manager's responsibility in this case is to discuss his account structure with the System Manager and System Supervisor prior to implementation. This will preclude any possibility of exceeding the directory space limitations.

NOTE

You may be unable to define the maximum number of groups permitted. The directory can be configured for a maximum of 96 groups, but the names of these groups are arranged alphabetically in a tree structure. If you assign two group names, and then create many groups whose names fall alphabetically between the original two group names, a branch of the tree structure may become full and permit no more entries. For best results, create groups in alphabetical order.

The Account Manager has the following commands available for his use:

:ALTGROUP	
:ALTUSER	
:ALTVSET	(Must have CV capability)
:LISTACCT	(For his account only)
:LISTGROUP	
:LISTUSER	
:NEWGROUP	
:NEWUSER	

:NEWVSET	(Must have CV capability)
:PURGEGROUP	
:PURGEUSER	
:PURGEVSET	(Must have CV capability)
:REPORT	(For his account only)
:SETCATALOG	
:SHOWCATALOG	

DEFINING GROUPS

To create a new group, the Account Manager logs onto the system and enters the :NEWGROUP command and any optional parameters.

Suppose that Account Manager Tom Smith has three programmers working for him. Jim Johnson is developing a parts inventory system. Mary Mason is developing a parts processing system. Bob Blake is developing an order processing system. Bob Blake also is responsible for maintaining and executing the current employee salary data base, which has just been converted from another computer system. Each programmer desires an individual group in order to save his developmental programs and files. Tom Smith consults his programmers and the System Manager. He proposes that four groups will be created, with three users. He charts the structure and the various attributes as follows:

GROUPS		USERS	
groupname	JOHNSON	*username*	JIM
filespace	10000 sectors	*password*	J2DEV
(*)		*homegroup*	JOHNSON
		(*)	
groupname	MASON	*username*	MARY
filespace	10000 sectors	*password*	AQUARIUS
(*)		*homegroup*	MASON
		(*)	
groupname	BLAKE	*username*	BOB
filespace	10000 sectors	*password*	ME98
(*)		*homegroup*	BLAKE
		localattribute	184
		(*)	
groupname	PAYROLL		
filespace	Unlimited		
password	MANYBUKS		
capabilitylist	BA		
(*)			

*All other attributes are the defaults assigned by the account.

Tom then logs onto the system and carries out the group creations. He enters:

:NEWGROUP PAYROLL ;PASS=MANYBUKS ;CAP=BA

In a similar manner, Tom defines groups BLAKE, MASON, and JOHNSON. As an example, the group structure created by the Account Manager for the PAYROLL group is:

groupname	PAYROLL
password	MANYBUKS
filespace	Unlimited
cpu	Unlimited time
connect	Unlimited time
capabilitylist	BA
fileaccess	(R, A, W, L, X, S: GU)

DEFINING USERS

Following the previous example, the Account Manager Tom Smith assigns new users to the previously created group structures. He enters:

```
:NEWUSER BOB; PASS=ME98; HOME=BLAKE; LOCATTR=184
```

Tom defines the other users, JIM and MARY, in the same way. As an example, the user structure created in the above :NEWUSER command for user BOB is:

username	BOB
password	ME98
capabilitylist	SF, ND, IA, BA
subqueuename	CS
localattribute	184
homegroup	BLAKE

ALTERING ACCOUNT ATTRIBUTES

The Account Manager can change one or more attributes of users or groups within his own account. The Account Manager, Tom Smith, desires to have a password (T1X2Z) on his own user name. Tom needs to prevent unauthorized entry to the Account Manager user's structure. He enters:

```
:ALTUSER TOM; PASS=T1X2Z
```

It should be noted here that the password is not in effect for Tom until he terminates the current session. The next logon to the system, however, will require the password.

ACCOUNT DISCUSSION

The account SMITH as defined by the Account Manager, Tom Smith, is now fully created. Figure 5-1 shows the basic structure within this account.

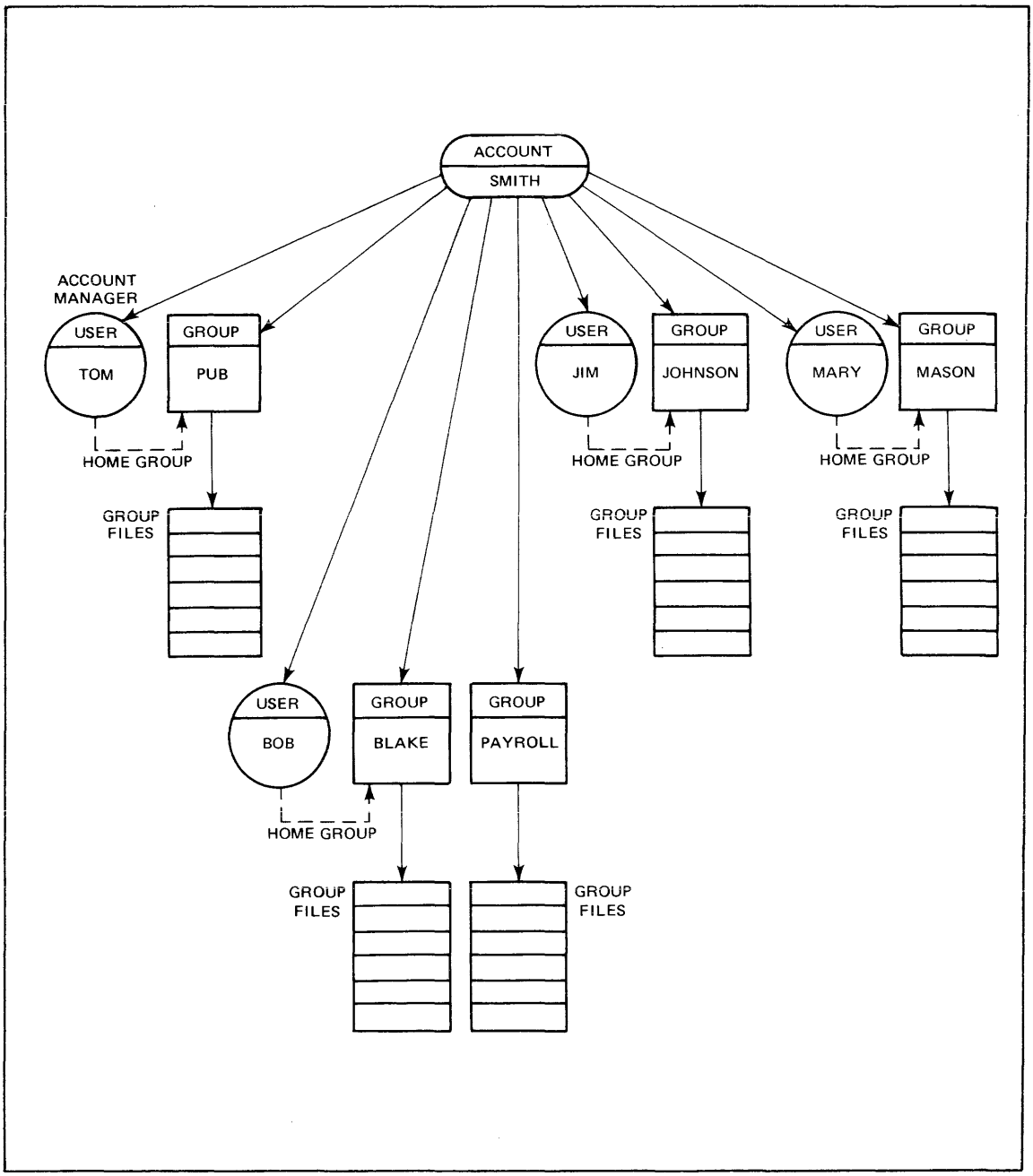


Figure 5-1. Sample Account Structure

Notice that each group is empty of files. Since neither Tom nor any other user has had the opportunity to create files, there is really no file space for those files as yet. Note also that the PAYROLL group has no user assigned. As stated previously, Bob Blake is responsible for maintaining and executing the payroll system. He also is responsible for development of an order processing system. For his development work, he has been assigned a homegroup of BLAKE. This does not prevent him from using the group PAYROLL as a *logon* group. A home group assignment is primarily for the convenience of the user, so that he will not be required to enter the name of the group desired each time he logs onto the system. Also, a user is not required to enter the password for his home group when logging on.

If Mary Mason desired to log on to the system and read a file belonging to the group JOHNSON, she would log on as follows:

```
:HELLO MARY.SMITH, JOHNSON
```

When the group JOHNSON is specified in the logon, it overrides the default home group assignment made by the Account Manager. Provided that Jim Johnson has not placed a lockword on the file that Mary Mason desires to read, she will be allowed access to that file.

In looking back at the account structure, you will notice that there is a password, MANYBUKS, assigned to the group PAYROLL. The reason for the password is to deny access to the PAYROLL group for other users within the account. In addition, since it is Bob Blake's responsibility to ensure the security of the payroll group and files, Bob has converted the payroll program and inserted a call to the WHO intrinsic. (This intrinsic is discussed in the *MPE Intrinsics Reference Manual.*) This conversion of the program has the purpose of checking the local attribute of the user running the program. Within the program, it specifically checks to see that the calling user has a local attribute of 184. In this way, a further check on the proper user is made prior to actual execution of the program. Looking back at the example where the user BOB was created, you will notice that a local attribute of 184 was assigned at that time.

Bob now moves the programs and data files into the PAYROLL group. Then, in batch mode, he assigns a lockword on the program file PAYPROG, using the :RENAME command as follows:

```
:RENAME PAYPROG, PAYPROG/NOWDOIT
```

This will provide further security for execution of the payroll program. In addition, he places a lockword on the data file and alters his program to be able to open the data file.

Suppose now that Bob Blake must run the new payroll for the current pay period. Bob gathers the new data (on cards), places his special logon and logoff cards in the deck, and places the whole deck in the card reader. He also will make arrangements for the blank checks to be placed in the high speed line printer at the appropriate time. He commences the job as follows:

```
:JOB PAYROLL, BOB/ME98.SMITH, PAYROLL/MANYBUKS
:RUN PAYPROG/NOWDOIT
   (Data cards containing latest employee information)
:EOJ
```

Now consider the various security checks that are required in order to execute the payroll program. The system first checks BOB's user password against the password that is being input. Then the system checks for a proper account name, SMITH. Then it checks the group password against the password which is being input. After being passed on all these checks, the user BOB is finally able to achieve logon. The next command calls for execution of the PAYPROG program. The system now checks for the existence of the program and then validates the password that is being input. At this time, the program is brought into execution, having passed every system security check placed by the Account Manager and by the user BOB. Now the program itself issues a call to the WHO intrinsic and checks the local attribute to ensure that the proper user is executing the program. The program then opens the old data file and commences to issue new pay checks for the period defined.

The above example shows how some of the various attributes assigned by the Account Manager and the user can be used to ensure a high level of privacy within an account. Note that the Account Manager Tom Smith has knowledge of the user's password, the group's password, and the user's local attributes. The Account Manager could also obtain information on the lockwords through use of the :LISTF command, with certain optional parameters allowed for Account Managers. It is then apparent that the Account Manager, as well as the user BOB, could access the PAYROLL group and execute the payroll program. Remember that the Account Manager can gain access to any file within the account, with *one* exception: an Account Manager cannot gain access to files with negative file codes unless he also has the Privileged Mode (PM) capability.

PURGING GROUPS AND USERS

The Account Manager user has the responsibility of removing users and groups which are no longer required within the account. He does this with the :PURGEGROUP and :PURGEUSER commands. If, for instance, a :PURGEGROUP command is entered, it will remove every file within the group and then remove the group from the account structure. If a file was somehow left in a state such that the operating system verified it to be in an open (busy) mode, then that file and the group itself would not be purged. Under those conditions, it is conceivable that the system would have to be restarted in order to clear the file, so that it and the group could be removed with another :PURGEGROUP command. Additionally, if no files are in use but a user is logged onto the system with the group used as a logon group, then all files will be removed — but not the group.

If a group is to be deleted from a private volume set, the volset parameter must be specified in the :PURGEGROUP command and a :MOUNT command must be entered for that volume set before the :PURGEGROUP command is entered. If the volume is already mounted due to an explicit or implicit request by a user, then all files not in use for the group will be purged but the group itself will not be purged.

The :PURGEGROUP command is useful for removing all files within a group, especially if there are a great many files to be removed. For example, suppose Bob Blake has used all of his disc space within his group BLAKE. In order to continue his development schedule without requesting more disc space, he saves all of the present files on magnetic tape or serial disc using the :STORE command (defined in Section II). Then he requests the Account Manager to purge the BLAKE group. The Account Manager logs onto the system in Blake's home group (and this is important) and enters:

```
:PURGEGROUP BLAKE
GROUP BLAKE TO BE PURGED? (YES/NO) YES
IN USE: CAN'T BE PURGED (CIERR 724)
```

What has occurred is the removal of all of Bob Blake's files from within his group. However, his group structure still exists because the Account Manager was logged onto the system in the home group BLAKE at the time the command was entered. This is the desired effect, and the error indication can be ignored.

Suppose now that Jim Johnson leaves the company. The Account Manager Tom Smith hires Fred Free to replace Jim Johnson, Fred desires to have his own group and user name, rather than using Jim Johnson's identification. Tom creates a new group, FREE, and a user, FRED, with the same attributes as previously granted to Jim Johnson. Fred then used the FCOPY utility (described in the FCOPY/3000 Reference Manual) to copy the pertinent programs and files into his group from the JOHNSON group. On completion of the transfers, Fred notifies Tom Smith that the group and *username* identified with Jim Johnson are no longer required. Tom logs onto the system and enters:

```
:PURGEGROUP JOHNSON
GROUP JOHNSON TO BE PURGED (YES/NO) YES
:
```

At this point, all files within the Johnson group have been removed, and the group structure JOHNSON has also been removed. However, the user name JIM and its attributes still exist within the account. Tom must issue a specific command to remove the user JIM from the account structure. He enters:

```
:PURGEUSER JIM
USER JIM TO BE PURGED? (YES/NO) YES
:
```

At this time, the user name JIM and all its attributes and any associated UDCs will be removed from the account structure. Note that, unlike the :PURGEGROUP command, if someone had been logged onto the system with the *username* JIM under the account SMITH, that user would have prevented the username JIM from being removed. The Account Manager would have to wait until the person using that *username* logs off.

ACCOUNT LIBRARIAN/GROUP LIBRARIAN

The Account Librarian (AL) and Group Librarian (GL) attributes are assigned to the Account Manager of a newly created account. The Account Manager can, in turn, pass on the AL and GL capabilities to whomever he chooses.

The Account Librarian (AL) attribute is normally assigned to a user in order to grant that user special file access for maintenance of certain files within the account. It is assigned to a user so that he may create and purge programs and account SL segments within the PUB group (or any group) of the account. Those files are usually accessible by any user of the account, but can only be modified or deleted by the user with the Account Librarian (AL) attribute.

The Group Librarian (GL) attribute is assigned to a user for the same reasons as an Account Librarian except that the Group Librarian attribute is restricted to the user's home group.

Basically, both attributes are meant to be used to control the degree of file access that standard users may attain. AL and GL do not give a user special command capability as do OP, AM, or SM.

DISPLAYING ACCOUNTING INFORMATION

ACCOUNT/GROUP/USER ATTRIBUTES

An Account Manager can list the attributes for his own account by using the :LISTACCT command. Also, by using the :LISTGROUP and :LISTUSER commands, he can list the attributes for one or all groups in his account, or for one or all users in his account.

Each listing appears as an octal dump showing one or more entries. Each entry is headed by A= (for accounts), G= (for groups) or U= (for users), followed by the appropriate name, and three or four lines of octal coded information. Examples and decoding information are given in Appendix D of this manual.

ACCOUNT/GROUP REPORTS

The accounting information for the log-on account, or for any or all groups in that account, can be extracted and displayed, showing counts and limits for permanent file space (in disc sectors), central processor time (in seconds), and session connect-time (in minutes).

The accounting information, requested with the :REPORT command, can be used for billing or for simply obtaining an overview of system usage on an account/group basis.

The :REPORT command is an expanded version of the command that is available to the standard user for displaying the total accounting information for his log-on account and group; this is discussed in Section IV and in the *MPE Commands Reference Manual*. For any particular resource, the sum of all group counts within any account always equals the total count accumulated by the account.

BACKING UP DISC FILES ON TAPE OR SERIAL DISC

Account Manager Users have the ability to dump and retrieve any files in their account by using the :STORE and :RESTORE commands. These commands are defined in Section II.

DEFINING UDC COMMANDS

A user-defined command or UDC is a procedure built from one or more standard MPE commands. When you combine several MPE commands into a single procedure and assign a name to that procedure, you create a UDC. In this way it is possible to enter the UDC name and cause several MPE commands to be executed. UDCs defined by the account manager may be known to and available to all users in the account.

User-defined commands may also be defined by the System Manager for use by all system users, and by individual users for private use. When identically named UDCs occur at different levels, the user level UDC takes precedence over the account level UDC which, in turn, takes precedence over system level UDCs.

UDC SYNTAX

The format of a user-defined command is fully described in Section III of the *MPE Commands Reference Manual.*

The Editor can be used to modify commands stored in a file, if that file is not being accessed. Note, however, that a UDC file, once entered into a catalog with the :SETCATALOG command, cannot be modified and kept under the same name with the Editor. You must first release the file from the UDC catalog by issuing the :SETCATALOG command with no parameters (which releases all files in the UDC catalog).

Establishing a UDC Catalog

The :SETCATALOG command is used to inform MPE that a specified file name contains user defined commands. The Command Interpreter then searches the file and establishes a directory for each command contained in the file. The file name is stored in a directory (COMMAND.PUB.SYS) of all UDC users. A user must have lock access specified for the group in which the file is located to use the :SETCATALOG command. Note that the default file access for the PUB group gives lock access only to users with AL, GL, or home group (GU) capability.

The UDC directory, COMMAND.PUB.SYS, must exist for the :SETCATALOG command to properly execute. If this file does not exist on your system, you must build it with a record size of 20.

If you want to have the file named AMUDC1 searched by the Command Interpreter, and have the user-defined commands contained therein entered into a catalog, enter:

```
:SETCATALOG AMUDC1;ACCOUNT
```

Note that once a UDC file is entered into a catalog with the :SETCATALOG command, that file cannot be purged with the :PURGE command or modified and kept under the same name with the Editor. Any attempt to do either of these two operations will result in an error message (Exclusive Violation).

In order to purge or modify such a file it must be removed from the catalog by issuing the :SETCATALOG command with no file names, as in:

```
:SETCATALOG;ACCOUNT
```

The effect is that all UDC files are removed from the catalog (the files are not purged, merely deleted from the catalog). This will take effect only after all users (except yourself) that had been accessing the file have logged off or issued a :SETCATALOG themselves. To re-enter UDC files into the catalog, re-enter the :SETCATALOG command for these files.

LISTING UDCs

To list the contents of a UDC file catalog, enter:

```
:SHOWCATALOG
```

MPE responds by providing a list of UDCs by assigned name and level (USER, ACCOUNT, or SYSTEM).

SYSTEM SUPERVISOR CAPABILITIES

SECTION VI

The System Supervisor (OP) capability allows a user to control the general operation of the system by:

- Reconfiguring the system (SYSDUMP).
- Storing all user files on the system offline on a magnetic tape or serial disc.
- Creating magnetic tape or serial disc files for backing up and/or modifying the system.
- Displaying certain system information for the purpose of determining the optimal performance.
- Allocating/deallocating programs in a loaded state.
- Exercising scheduling control over processes in order to obtain the best system performance.
- Managing the system log files through the logging facility.
- Establishing the number of user logging processes allowed on the system and the number of users per logging process.

The System Supervisor (OP) capability differs from the System Manager (SM) capability (which is mostly concerned with the accounting system) in that the System Supervisor is generally responsible for:

- Managing the entire operating system software on a day-to-day basis.
- Ensuring that file offline storage for all users is accomplished on a regular basis.

In order to perform these functions, the System Supervisor uses the following commands:

:ALLOCATE
:DEALLOCATE
:RESTORE (For all files in system)
:RESUMELOG
:SHOWLOG
:SHOWQ
:STORE (For all files in system)
:SWITCHLOG
:SYSDUMP
:TUNE

Additionally, the System Supervisor is responsible for the system initialization procedure when modification to the operating system is to be made during start-up procedure. (The start-up procedure normally is the Console Operator's responsibility — except when operating system modifications are to be made.)

Hewlett-Packard furnishes MPE as a set of prepared program files on serial storage media. In this initial configuration performed at the factory by Hewlett-Packard the following are included:

- One user with System Manager and System Supervisor capabilities (plus all other standard and optional capabilities from the MPE capability set). This user is identified by the username MANAGER.

- One user with all the capabilities of the MANAGER user including the System Manager (SM) capability. This user is identified by the username FIELD.
- Two system accounts. The accounts are SYS and SUPPORT. The MANAGER user is associated with the SYS account and the FIELD user is associated with the SUPPORT account.
- One public group (PUB) belonging to the system account SYS.
- Numerous groups which are associated with the account SUPPORT. There is one group for each supported software product. Each groupname is defined as HP XXXXX where XXXXX is the Hewlett-Packard product number of the appropriate software product.

In addition, magnetic tapes and serial discs are provided to load pertinent subsystems (may be on the same magnetic tape as the operating system), and perform stand-alone cpu and non-cpu diagnostics.

SYSTEM CHANGES AND FILE STORAGE

Once the system is started (by the Console Operator) and some accounting structure and files exist, the System Supervisor can call the MPE Configurator program :SYSDUMP to:

1. Perform a system and file backup to magnetic tape, serial disc, or cartridge tape unit, by:
   a. Copying the operating system only.
   b. Copying the operating system plus the system directory (accounting structure).
   c. Copying the operating system, accounting structure, and all files in a given fileset description that have been altered since a given date.
   d. Copying the operating system, accounting structure, and all files on the system.

2. Change the operating system, as well as any option provided in 1b through 1d above.

NOTE

Duplicating the running system on magnetic tape, serial disc, or cartridge tape unit, is the only way that you can provide backup. If no such backup exists and a system failure occurs, you must execute one of the supported utilities in order to retrieve files. You also must contact Hewlett-Packard in order to obtain another copy of the operating system. For these reasons, it is highly recommended that a backup of the system and of all the files on the system be performed on a regular basis.

Regardless of whether the Configurator (:SYSDUMP) is used to reconfigure MPE or to make an identical copy of it, the running system is not altered; any changes made apply only to the copy of the system written to magnetic tape, serial disc, or cartridge tape unit.

You will need certain background information when using the Configurator (:SYSDUMP) to modify the operating system. The information you will need depends on the types of changes to be made. Changes are made through an interactive dialog in session mode between you and the Configurator. The following paragraphs and Appendix C of this manual provide background information for those portions of the dialog which are not self-evident.

CHANGING INPUT/OUTPUT DEVICE CONFIGURATION

Every physical input/output device in the system is identified by a unique *logical device number,* ranging from 1 to 255. Input/output configuration consists of specifying this number and various other characteristics for each such device. Some of these characteristics, such as Device Reference Table (DRT) entry number and device unit number, described below, are determined by physical hardware connections made prior to system generation. Other characteristics, such as whether a device is interactive or duplicative, whether it can accept jobs and sessions, and the device class to which it belongs, are user options. When you are deleting or re-specifying devices already on the system, you can determine the characteristics of these devices by requesting a device characteristics listing during the re-configuration process. When you are adding a new device, you must know the hardware-dependent characteristics of the device and must also carefully determine those characteristics that are user options, as noted below. The characteristics that must be specified for each device are:

- Logical Device Number (LDN)

  The logical device number is the value by which the MPE File Management System recognizes a particular device. For each device, this is a unique number ranging from 1 to 255. You may assign logical devices in any way you choose (except that the system disc must be logical device number one). For example, assume there will be only three devices to configure on the system. They are: a disc, a magnetic tape, and a terminal. You could arbitrarily assign logical devices as follows:

Logical Device	Device Type
1 (Must be system disc)	Disc
4	Magnetic Tape
152	Terminal

Normally, you would assign logical devices in ascending order.

- Device Reference Table (DRT) Entry Number

  Every device on the system is connected to a device controller. A particular controller may serve more than one device of the same type. There is an entry in the DRT in main memory that contains information about the driver and interrupt programs that serve an individual controller and its devices. Because each DRT entry is four words long, the size of the DRT, in words, is four times the total number of controllers.

  The location of the DRT is given by absolute locations 8 and 9, which hold the memory bank of the DRT and the address of DRT 0, respectively. For systems with a limit of 127 DRT entries (i.e. all systems except the Series 64) the DRT will be located in bank 0, starting at address 0. For the Series 64, the DRT is a part of linked memory in a region which is permanently reserved and frozen, so that the DRT does not move during execution of MPE.

DRT entry numbers 0, 1, and 2 do not exist. However, the memory locations where DRT 0, 1, and 2 would normally be located are used by the operating system. DRT 3 is always the system clock and is never configured into the system I/O configuration table.

When re-configuring the system, you need to know the highest DRT entry number that can be assigned to a device. This value may not exceed 127, or 511 for the Series 64 computer.

You also need to know the DRT entry number of any device to be added or deleted. On the Series II/III, DRT numbers range from 3 to 127. The number is determined by a set of jumpers and switches on the device controller board.

DRT numbers on the Series 30, 33, 40, and 44 range from 8 to 127. This number is composed (in binary) of a 4-bit channel number followed by a 3-bit device number. For example, a device on channel 5, device number 3, would be on DRT number %53.

DRT numbers on the Series 64 range from 8 to 511. This number is composed (in binary) of a 2-bit IMB number, followed by a 4-bit channel number, followed by a 3-bit device number. For example, a device on IMB 1, channel 2, device 1 would be on DRT number %121. Devices may not be configured on channel 0 of any IMB.

- Unit Numbers

  When a controller services only one device, that device is assigned a unit number of 0 (recognized by the associated driver). When the controller serves more than one device, each device is assigned a unique unit number (with respect to that controller) to distinguish it from others connected to the same controller. The unit number of any device is a hardware-dependent characteristic determined when the device is physically connected to its controller. The value ranges from 0 to a maximum number determined by the type of device controller.

- Software Channel Number

  The input/output (I/O) system allows multiple controllers on a single hardware channel. The system will transfer data on only one controller per software channel at one time, and a queue is established on each channel for the I/O pending on other controllers. All controllers with the same channel number are assumed by MPE to be on the same software channel, even though physically they need not be on the same hardware channel to share the same software channel number. Thus you may deliberately assign several controllers to the same channel number as a method of preventing simultaneous transfers to a group of controllers. A software channel number of zero (0) means do not put the device on a software channel.

- Option to Accept Job and Session Input Stream (J)

  You can optionally specify that this device can accept the commands :JOB or :HELLO, and thus serve as a job/session input device.

  For the purpose of this option, a :JOB is an input stream which has been completely defined prior to any input on a device. A session (:HELLO command) is an input stream of data defined as it is input by a user.

  Note that this option provides for automatic recognition of the device by the operating system for either jobs or sessions. It should be noted also that sessions cannot be started unless this option as well as the *Interactive* option (defined below) is specified.

- Option to Accept Data from a Device (A)

  The purpose of this option is to allow a job or session to acquire an auxiliary input device different from $STDIN without operator intervention. In order to do this, the auxiliary device must be configured as a :DATA accepting device. For example, a session on a terminal could read cards from a card reader if the reader is configured as a Data Accepting device and the user entered the appropriate :DATA card as the first card of the data.

  This option provides for automatic recognition of the device by the operating system. In other words, as in the example above, the :DATA card will be read by the operating system when the device is made ready. If the device is also spooled, the data also will be input immediately to a disc device file.

- Interactive Option (I)

  Interactiveness is a characteristic of a device that requires human intervention for all input operations. This quality is necessary to establish the man/machine dialog required to support a session.

  In this regard, the option to accept job and session input streams (defined above) must also be specified. Note that an interactive device can support both jobs and sessions. When an interactive device is used as a :JOB device, the operating system assumes that there is a pre-defined input stream (i.e., there is no immediate human intervention available for error recovery). If a user is manually inserting the input stream and makes an error, the operating system will terminate the job.

- Duplicative Option (D)

  A device is configured as duplicative when all input operations can be echoed to a corresponding display without intervention by the operating system software. In general, devices which are interactive are also duplicative. The hardware design of the system is such that all input to a terminal device will be echoed back by the hardware to the display portion of the terminal. Thus the user can see what has been input to the system software. A device is configured as non-duplicative when it cannot accept this hardware echoe to its corresponding display and it does not locally echo the input. If a device has been configured as non-duplicative, the command interpreter and several subsystems will "echo" back user input. An example of a device which is configured as non-duplicative is a card reader.

  Some non-standard terminals have a self contained echo feature. If there is no switch available on the terminal to turn off this capability, the session logon will appear to be garbled because the system hardware is echoing back to the terminal and the terminal is echoing to itself. Once logon is attained, the user can depress the escape and semi-colon (ESC and ;) keys which will stop the system hardware from echoing the input back to the terminal.

  If a device is configured as a job/session device and is also an interactive device, then, in general, the duplicative option should be specified.

- Enable Seekahead (E)

  This option is available to make more efficient use of seek time for the configured device type 0 or 1, a disc drive. If the disc drive is to be configured as more than one unit, for example, subtype 4 or 5, a 7905 with a removable or fixed platter, lookahead seeks should not be enabled. (Answer NO). If the disc is to be configured as one unit, answer YES to enable seekahead.

  NOTE: For Series II/III only.

- Spooling Option (S)

  When the operating system is started up, certain devices can be started automatically by the spooling facility in the spooled state. The only devices which can be spooled are:

  Input Spooling :JOB and/or :DATA accepting card readers and reel-to-reel magnetic tapes.

  Output Spooling Line printers and plotters.

  NOTE

  The card reader/punch may be spooled as an input device or as an output device, but not both at the same time.

- Corresponding Output Device

  If the device can be used as a job/session input device, you must specify (by logical device number or device class name) a device that will be recognized as the corresponding job/session list device. That is, all input read from the job/session input device is listed on that particular list device (or one of a set of devices, if a class name is specified). In the case of discs and output-only devices such as line printers, a zero should be entered.

- Driver Name

  The name of the driver associated with the device/controller is specified. For standard devices supported by Hewlett-Packard, appropriate driver names are found in Appendix A. For non-standard drivers supplied by the user, this is the name of the program file containing the driver; the name must contain from one to eight alphanumeric characters, beginning with a letter. If the driver name is preceded by an asterisk (*), the driver will permanently reside in main-memory.

- Device Classes

  The general class to which a device belongs may be specified. This enables a user to request a device by class name, such as any disc or any tape unit. These names are arbitrary, installation-dependent names that are left to the discretion of the System Supervisor. They consist of up to eight alphanumeric characters, beginning with a letter. A device can belong to more than one class, such as DISC and SPOOL.

  The classes DISC and SPOOL (if spooling is desired) are specifically required by MPE. DISC is the default device class for building files. SPOOL is the device class for designating "spooling discs"; spoolfiles will only be allocated on discs which are included in the special device class SPOOL. User files also may reside on spooling discs.

  The following classes must also be included if the User Logging Facility will be used with devices in a class other than DISC:

  TAPE Must refer to devices in a magnetic tape class.

  SDISC or CTAPE Must refer to devices in a serial disc class.

MPE defines seven categories of I/O devices. By convention, each device supported by MPE is assigned a device type number which places it in one of the seven categories. Each category uniquely confines the mode of access to operations consistent with the properties of devices in that category.

Device Types	Device Access Type (DAT)
0-6	Direct access (DA)
7	Foreign disc (FD)
8-15	Serial input (IN)
16-23	Concurrent input/output (I/O,C)
24-30	Non-concurrent input/output (I/O,NC)
31	Serial disc (SD)
32-39	Serial output (OUT)

The distinction between concurrent and non-concurrent is as follows:

concurrent — input and output operations are allowed in any order while the file is open. For example, terminals may be written to or read from in any order.

non-concurrent — input and output operations are allowed, but their order may be restricted. For example, you may not read from a magnetic tape immediately after writing to it, but you may backspace over the written record and then read it.

NOTE

This is to help you with "class changes."

A device class has a device class access type (DCAT) associated with it. The set of DCATs includes DATs as well as specific device types. DCATs can broaden the scope of a device class so that it may contain devices from more than one DAT. This concept may be used to restrict the mode of access of devices within a class to a subset of the operations that the devices themselves are capable of performing. This facility is effective only when a device is opened by class name; when it is opened by logical device number, the mode of access is determined only by its DAT.

The Configurator dialog for assigning devices to classes permits the installation to specify the preferred order of allocation of devices within a class.

The following rules apply when assigning devices to a class:

a. A class containing only removable, moving-head discs will be assigned a DCAT of direct access (DA) or serial disc (SD), depending on its specification when it is initially defined.

   Classes containing only direct-access devices, but containing one or more non-removable discs, will automatically be assigned to DA.

   Note: Class "DISC" *must* be a DCAT of DA.

b. A class containing one or more serial input DATs is automatically assigned a DCAT of serial input if the remainder of the DATs in the class are input/output (either concurrent, non-concurrent, or both). For example, a normal ASR 33 TTY and a high speed paper tape photo reader. The TTY is normally a concurrent input/output device while the photo reader is always a serial input device. If a class of PTIN is assigned to the photo reader and the TTY, then the class PTIN is assigned a DCAT serial input (IN) only.

c. A class containing one or more serial output DATs is automatically assigned a DCAT of serial output if the remainder of the DATs in the class are input/output (either concurrent, non-concurrent or both). For example, a card punch is always a serial output device (OUT). A magnetic tape is a non-concurrent input/output device (IN/OUT,NC). If a class of PUNCH is assigned to the card punch and the magnetic tape, then the class PUNCH will have a DCAT of serial output (OUT) only.

d. For a class containing only concurrent input/output DATs, the Configurator will prompt to ask whether the DCAT should be serial input, serial output, concurrent input/output, or non-concurrent input/output. For example, when adding a new terminal to the configuration, you will be asked for the type of DCAT assignment since terminals can be concurrent input/output devices (IN/OUT,C).

e. For a class containing only non-current input/output DATs, the Configurator will prompt to ask whether the DCAT should be serial input, serial output, or non-concurrent input/output. An example of this type would be when adding a new magnetic tape device to the configuration.

f. For a class containing a mixture of concurrent and non-concurrent input/output DATs, the Configurator will prompt to ask whether the DCAT should be serial input, serial output, or non-concurrent input/output.

g. A class containing any combination of DATs other than those mentioned above will be illegal. For example, a line printer is an output device (OUT). A card reader is an input device (IN). A class which would specify both devices for the class is illegal.

h. A class containing devices all of the same type is assigned a DCAT of that same device type.

AUTOMATICALLY ALLOCATING MAG TAPE UNITS: Installation and Use

Magnetic tape units may be automatically allocated by meeting *all* of the following conditions. If *any one* of the conditions is not met, the tape unit is allocated in the standard manner (via a 'REPLY' from the console).

1. The magnetic tape unit must be device type 24. The device subtype for a 7970B/E which is automatically allocated must be 8. The default subtype for a 7970B/E is subtype 0, and the unit will not auto-allocate. The subtype for an automatically allocated 7976A is subtype 9. The default subtype for 7976A is 1, and the unit will not auto-allocate.

   Note: Any existing allocated 7970B/E with subtype 15 should be reconfigured as subtype 8.

2. The tape unit must not be JOB or DATA accepting.

3. The tape drive must not already be allocated to another user.

4. The device specification, logical device number or device class, must be unique in order to specify only one device in the system.

   Note: If more than one logical device is in a class, any reference to this class name will require the operator to 'REPLY'. An example of an I/O configuration for the 7970B/E and 7976A follows:

   7970B/E (7976A)

LDEV	TYPE	SUBTYPE	MODE	CLASS NAME(S)
7	24	8 (9)		TAPE, DOUG
8	24	8 (9)		TAPE, BRENDA
9	24	0 (1)		TAPE, CAROL
10	24	0 (1)	JA	TAPE, SARA

   Action taken as result of above configuration via File equation.

DEVICE REFERENCE	ALLOCATION	REASON
DEV=7	auto	all conditions met
DEV=BRENDA	auto	all conditions met
DEV=TAPE	console	more than one device in class tape
DEV=9	console	subtype not = 8(9)
DEV=SARA	console	subtype not = 8(9) and JOB & DATA Accepting

5. If the FOPEN request is for a labeled tape, and the tape mounted is not labeled, an operator :REPLY is required to successfully open the tape file. Also, if the VOLID has not been previously specified in the FOPEN request or a :FILE command, the operator is is required to supply it (via :REPLY) the first time the tape is opened.

CHANGING SYSTEM TABLES AND QUEUE PARAMETERS

The sizes of various system tables, queues, and other values can be changed to permit you to make most efficient use of main memory and virtual memory. You can best determine what particular values are best for your installation by comparing the values supplied by Hewlett-Packard in the initial configuration with those later suggested by your own operational experience. A table of default values (those normally provided initially) appears in Appendix C. The System Supervisor should experiment with various parameters over a period of time in order to achieve best system performance.

Elements that can be changed are:

- The size of the following tables

  a. Code Segment Table (CST)
  The CST is one continuous main memory resident table. It is divided into the program and sharable areas by the system. Each entry is a 4 words in length. Experimentation with both the sharable and program areas is advised in order to conserve main memory allocation.

  b. Data Segment Table (DST)
  This table is main memory resident. Each entry is 4 words in length. Experimentation with size is advised in order to conserve main memory allocation.

  c. Process Control Block (PCB) Table
  This table is main memory resident. Each entry is 16 words in length. Experimentation with size is recommended.

NOTE

In all of the tables above, setting any one table with a size which is too small will cause the operating system to fail. Experimentation with sizes should be done with caution.

- The maximum number of entries in

  a. Input/Output Queue (IOQ)
  This queue is main memory resident. Each entry is 11 words in length. For each non-disc I/O request, an entry is allocated. If devices are spooled, at least 15 entries for each spooled output device and 10 entries for each spooled input device are needed. Experimentation is advised. However, the absolute minimum number of entries recommended is 20 regardless of the configuration; the maximum number is 256. Setting the number of entries too low for your configuration may cause the operating system to fail. Care should be used when experimenting with the number of entries.

  b. User Controller Process (UCOP) Request Queue
  This request queue resides in virtual memory. It is main memory resident only when needed by the system. Each entry in the queue is 2 words in length. Experimentation with size is advised. Setting the entry size to a low value can cause performance degradation of the operating system. Caution is advised when lowering the number of entries.

  c. Disc Request Table
  This table resides in main memory. Its structure is parallel to the IOQ table which holds the request parameters for pending disc I/O's. Each entry is 16 words in length. For the typical system, a minimum of 60 entries is recommended. The absolute minimum number of entries is 16, and the maximum is 255. The failure to allocate enough entries will result in System Failure 601.

(cont'd)

d. Swap Table
This request table is a memory resident data structure used to keep track of the segments referenced by each process. Each entry is 5 words in length. There is one entry allocated for each segment that is being used by the process. The suggested number of entries is 4 times the maximum number of PCB entries configured in the system. The maximum setting is 2048. Failure to allocate enough entries will result in a System Failure 602.

e. Primary Message Table
The Primary Message Table is used to buffer message content for pending messages transmitted between system processes using the memory resident message system during the execution of critical areas of system code. It is memory resident. Each entry is 5 words in length. The maximum number of entries is 255, with the suggested number of entries being 25. Failure to allocate enough entries will result in a System Failure 620.

f. Secondary Message Table
The Secondary Message Table is used to buffer message content for pending messages transmitted between system processes using the memory resident message system. The table is memory resident. Each entry is five words long. The maximum number of entries is 255 and the suggested setting is 25. Failure to allocate enough entries results in processes requiring this table to become impeded until entries become available, thus impacting system performance.

g. Special Request Table
This is a memory resident table used to temporarily buffer the parameters for segment expansions and to form the queue for devices waiting for a segment to arrive. Each entry is 5 words long. The maximum number of entries is 255 and the suggested setting is 25. Failure to allocate enough entries results in a System Failure 600.

- The number of buffers

a. Terminal Buffers in the system
Terminal buffers are main memory resident. Each buffer is 16 words long (30 characters — one word is used as a link). Set the number of buffers to about 5 times the actual number of terminals on the system.

NOTE: Terminal buffers in Series 30/33/40/44/64 systems are 32 words in length (60 characters).

b. System buffers in the system.
System buffers are main memory resident. Each system buffer entry is 129 words long. System buffers are used by the file management system, the input/output system and the data communications subsystems, to reduce the number of extra data segments required for any particular user. Experimentation with the number of entries may be done. The file system uses system buffers numbered above 8 (system buffers one through eight are not used by file system). The recommended number is 8 plus 1 for each MTS data communications line.

- The number of words on the Interrupt Control Stack (ICS)

  This stack is main memory resident. The system Dispatcher executes from this stack. Additionally, the input/output system uses this stack for certain processing. Experimentation with ICS size should be done with extreme caution. An ICS whose size is too small will cause the operating system to fail.

- Timer Request List (TRL)

  This lists consists of 4 words per entry. The list is main memory resident. In general, the system places requests in this list in order to *delay* a process or some function for a specified amount of time. If the list is configured too small, the operating system will fail. Care should be taken when reducing the number of entries.

- The size of the breakpoint table

  This is a disc resident table. Entries are variable in length. This table is used by the DEBUG facility to save the various breakpoint addresses for users. If you anticipate very little program debugging on your system, you may set the table size to some minimal value. The absolute minimum recommended for any system configuration is 24 entries.

- The number of user logging processes permitted.

  The number of logging processes permitted on the system affects the size of the logging identifier table. Although there are minor performance considerations, the flexibility of your logging requirements should be the determining factor. The number specified is the exact number of user logging processes permitted to run on the system. Normally one process per application, which enables one user logging file per application, is sufficient.

- The number of users permitted per process

  The number of users per process limits the number of users that can access a user logging file. Determine the maximum number of users that will need to access a logging file and enter that number.

- The contents of the message catalog

  This is the system data segment where MPE keeps its standard messages and error messages. The source file is CATALOG in the group PUB.SYS.

  See Appendix G for a description of the MPE message system.

CHANGING MISCELLANEOUS VALUES RELATING TO LOG-ON TIME, RIN'S AND JOBS ALLOWED IN SYSTEM AND EXECUTION

You can change

- The currently-assigned global resource identification numbers (RIN's).
- The number of RIN's available in the RIN pool.

- The maximum number of global RIN's available.
- The time a user is allowed to successfully complete logging-on to the system when initiating a session.
- The maximum number of jobs allowed in execution at one time.
- The maximum number of sessions allowed in execution at one time.

NOTE

The number of jobs and sessions allowed in execution at one time may be controlled by the console operator command :LIMIT within the maxima configured for the system. HIPRI jobs and sessions may execute regardless of the current limits, but if the total number of jobs and sessions attempting to execute exceeds the sum of the configured maxima, you will get a job overload type 4.

- The default central processor time limit for jobs.

CHANGING LOGGING CHARACTERISTICS

You can change

- The elements (types of entries) being logged in the system log file. This includes disabling or enabling the logging facility itself.
- The size of blocks in the log file.
- The size of the log file.

Note: See "Log Records" page 6-131.

CHANGING DISC ALLOCATION

You can alter the maximum number of disc sectors available for virtual memory. Note that these sectors are used for data-segment swapping only, since code segments are read directly from files and need not be written back to disc.

The maximum number of disc sectors available for the system directory also can be changed. This directory contains the addresses of all files, plus the accounting structure for all accounts, groups, and users in the system.

The most efficient values for virtual memory and system directory sizes depend on the installation and its use. See Appendix E.

The maximum number of spoolfile kilosectors and extent sizes also can be changed. Experimentation with the sizes is recommended. If a value is too small, the spooling facility will stop. Care should be exercised in setting a smaller value.

CHANGING SCHEDULING QUEUE

All processes competing for access to the central processor access it through the MPE Scheduling Queue. You may change scheduling parameters C, D, and E base, C, D and E limits, and C, D, and E min and max via the :TUNE command.

Care should be taken in altering any of the parameters as system attempts to automatically self-tune and thus any additional changes may greatly affect the system performance.

For a description of the :TUNE command and scheduling queues, see page 2-65.

CHANGING SEGMENT LIMITS

You can change the limits on code and data segments, as follows:

1. Maximum number of concurrently running programs.
2. Maximum number of words allowed in any code segment. A useful rule of thumb is to set the maximum size to 5% of available memory but in no case less than 4096 words.
3. Maximum number of code segments per program file.
4. Maximum number of words allowed in any user process' stack. Obviously, the maximum number of words in a stack is highly installation dependent. However, through experience it has been found that installations/users are using:

Max Stack Size (words)	Environment
7000- 8000	School/students
12000-16000	Scientific/Engineers
26000-31232	Commercial

   Note that smaller stack sizes increase multiprogramming performance.

5. Maximum number of words allowed in any extra data segment. A useful rule of thumb is to use 20% of the maximum number of words allowed in the user process stack. (See 4 above.)
6. Maximum number of data segments per process.
7. Default number of words initially assigned for a user's stack (Z-Q) area (when the user specified no value at program execution time). We recommended keeping this value as low as possible. The operating system will allow for dynamic increases of the user's stack up to the maximum allowable, if the program is PREPed with a MAXDATA parameter specified (see 4 above). In any case, the minimum recommended for any configuration is 1200 words.

Generally-recommended values for initial configurations can be found in Appendix C. Over a period of time, the System Supervisor should develop the best values for his installation through experimentation.

CHANGING SYSTEM PROGRAMS AND SEGMENTS IN THE SYSTEM LIBRARY

You can replace system program files. You also can delete, add, and replace code segments in the System Library. During configuration, you can request a list of all code segments currently in the System Library. This list can be useful in finding the logical segment number of a system segment which returned an error to a user or user program.

USING THE :SYSDUMP COMMAND

The :SYSDUMP command can be entered at any time during a session. Prior to entering the command, however, a file reference to a serial storage device must be made. (If a disc drive is used, it must be an HP 7906, HP 7920, HP 7925, or HP 7933.) A file reference to a list device also can be made (optional).

NOTE

With Series 30/33/40/44/64 systems, the disc drive may be an HP 7902 or 9895 flexible disc drive. With Series 40SX systems, and Series 40/44 systems with less than 128 megabytes of disc space, the integrated cartridge tape unit may be used as a backup medium.

To begin dialog with the Configurator, the commands below could be used.

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

When the :SYSDUMP command is entered, the Configurator begins an interactive dialog with you by asking if you want to make any configuration changes. If the answer is NO, the Configurator copies the running system to the specified device (TAPE in this case). If the answer is YES, the Configurator continues its dialog.

NOTE

If $NULL is specified as the output file, SYSDUMP will not write to any device. This may be useful in displaying the current system configuration. "DUMPTAPE" cannot be used as a formal file designator. (See discussion on IMPLICIT :FILE COMMANDS FOR SUBSYSTEMS in Section II of the MPE Commands Reference Manual (30000-90009)).

Notes Concerning Data Communications Products

There are special considerations required when configuring one or more Data Communications Subsystems into your system.

Data Communications products currently available are as follows:

- DSN/Distributed Systems (DSN/DS)(Series II/III or Series 30/33/40/44/64)
- DSN/Remote Job Entry (DSN/RJE)(often called "the 2780/3780 Emulator"). (Series II/III or Series 30/33/40/44/64)

- DSN/Multi-leaving Remote Job Entry (DSN/MRJE)(Series II/III or Series 30/33/40/44/64)
- DSN/Multipoint Terminal Software (DSN/MTS)(Series II/III or Series 30/33/40/44/64)
- DSN/Interactive Mainframe Facility (DSN/IMF)(Series II/III or Series 30/33/40/44/64)

For each of the above, the configurable items are:

DSN/Distributed Systems

1. A Hardwired Serial Interface (HSI), driver name CSHBSC0. An HSI is used for coaxial connection between computers to transmit data at very high rates of speed. (Only available on Series II/III). The HSI has up to four ports (HSI channels). Each port is configured with a different logical device number and corresponding port mask (note that only one port may be used at any one time). This applies only to Series II/III.

2. A Synchronous Single-Line Controller (SSLC), driver name CSSBSC0, or an Intelligent Network Processor (DSN/INP), driver name IOINP0. Both the SSLC and DSN/INP can be used with modems or with hardwire cables.

   (Note that an SSLC is available only for use on Series II/III Systems.)

3. A Communications Driver; name IODS0. For an HSI, the driver is configured up to four times, once for each HSI channel. For an SSLC or DSN/INP, the driver is configured only once.

4. One or more pseudo terminals; driver name IODSTRM0. Because only one port on the HSI can be opened at a time, only one block of pseudo terminals needs to be configured. The block is automatically reallocated to a port when the port is opened. Thus one block of terminals serves all ports.

DSN/Remote Job Entry

One SSLC for each line to the remote system; (driver name CSSBSC0) or one DSN/INP for each line (driver name IOINP0).

DSN/Multi-leaving Remote Job Entry

1. One SSLC for each line to the host system, (driver name CSSBSC0); or one DSN/INP (driver name IOINP0).

2. One additional driver (driver name CSSMRJE0).

3. Two pseudo line monitors for each line (driver names IOMRJE0 and IOMRJE1).

4. One pseudo terminal for each line (driver name IOMCONS0).

5. At least one (optionally, up to seven) pseudo line printer for each line (driver name IOMPNLP0).

6. Optionally up to seven (none required) pseudo card punches for each line (driver name IOMPNLP0).

7. At least one (optionally, up to seven) pseudo card readers for each line (driver name IOMRDR0).

DSN/Multipoint Terminal Software

1. One SSLC for each line (driver name CSSBSC0); or one DSN/INP (driver name IOINP0).

2. One additional driver (driver name CSSBSC1).

3. One multipoint supervisor for each controller (driver name IOMPS0).

4. One or more multipoint terminals for each controller (driver name IOMPTRM0).

Configuration requirements for all data communications devices and pseudo devices are summarized in Appendices A and B. Refer to the tables in Appendix B when you encounter configuration prompts that pertain to any of the following:

- HSI--Table B-21.
- DSN/INP--Table B-25.
- SSLC--Table B-18.
- DSN/Distributed Systems Communications Driver--Table B-19.
- DSN/Distributed Systems Pseudo Terminals--Table B-20.
- DSN/Multi-leaving Remote Job Entry Pseudo Devices--Table B-22.
- DSN/Multipoint Terminal Software Devices--Tables B-23 and B-24.

DSN/Interactive Mainframe Facility (DSN/IMF) -- Table B-26.

1. One Intelligent Network Processor (DSN/INP) for each modem line to the host. Specify device type 17, unit number 0, subtype 1, driver IOINP0.

2. One IMF pseudo device for each DSN/IMP that is to be used for DSN/IMF. Specify device type 22, unit number 0, subtype 1, driver IOM3270.

SYSTEM MODIFICATION

The Configurator's output consists of questions and statements. The content of the question indicates the type of answer required. To those questions requiring a simple affirmative or negative answer, respond with YES (or Y) or NO (or N, or a carriage return). Other questions contain values followed by a question mark; they normally specify an existing parameter value or, the number of entries in use (min) followed by the existing maximum (max), and ask whether you want to change it. To retain the specified value, enter a carriage return. To change the value, enter the new value which, in the min/max case, is the new maximum value. In any case, you must always conclude an entry with a carriage return to transmit the entry to MPE.

Step No.	Dialog
1	ANY CHANGES? To prepare for changes, enter YES. To omit changes and skip to Step 148, enter NO.
2	SYSTEM ID = HP 32002 V. <UU> . <FF>? (HP 32033V . < UU> . < FF> on Series 30/33/40/44/64 systems) V is the current MPE version, UU is the present update-level number, and FF is the fix-level number. To prepare for updating software for a new fix level, enter the new fix-level digits (FF). (These digits indicate the latest system fix provided by Hewlett-Packard.) Otherwise, enter a carriage return.
3	MEMORY SIZE = <XXXX> ? XXX denotes the present size of main memory. To indicate the size of main memory for the system for which MPE is being configured, enter one of the following values: 128, 160, 192, 224, 256, 384, 512, 768, 1024, 1280, 1536, 1792, 2048, or 4096. This indicates the memory size in a multiple of 1024 words. NOTE: The series type (II, III, or 30/33/40/44/64 determines which of these memory sizes are available. To retain the present memory size, enter a carriage return.
4	I/O CONFIGURATION CHANGES? To prepare for addition or deletion of input/output devices, enter YES. To maintain the same input/output device configuration and skip to Step 74, enter NO.

Step No.	Dialog

5 LIST I/O DEVICES?

Enter YES for a list of I/O devices. The format is:

```
LOG DRT U C T SUB     TERM     REC     OUTPUT  MODE  DRIVER    DEVICE
DEV  #  N H Y TYPE TYPE SPEED WIDTH   DEV             NAME      CLASSES
        I A P
        T N E
```

The headings denote the following:

LOG DEV	Logical device number
DRT #	Hardware device address (Device Reference Table number)
UNIT #	Hardware unit number of device on its controller.
CHAN	Software channel number of device on its controller.
TYPE	Device type
SUBTYPE	Device subtype
TERM TYPE	Terminal type
TERM SPEED	Terminal speed
REC WIDTH	Record width in decimal words
OUTPUT DEV	Device class name or device ldn
MODE	E = Enable seekahead J = Accept jobs A = Accept data I = Interactive device D = Duplicative device S = Spooled device
DRIVER NAME	Driver name (See Appendix B for driver data)

To suppress this listing, enter NO.

NOTE

The prompt in Step 6 appears only if a communications interface (SSLC, INP and/or HSI) is currently configured into the system.

Step No.	Dialog

6 LIST CS DEVICES?

To list the characteristics of all Communications Systems devices currently assigned to the system, enter YES. The format of the output is:

```
LDN PM PRT LCL TC  RCV   LCL   CON  MODE   TRANSMIT  TM BUFFER D DRIVER
           MOD    TMOUT TMOUT TMOUT          SPEED       SIZE  C OPTIONS
```

If you have a switched device, such as those that are connected through a dial-up telephone line, then you will receive the additional output:

```
LDN CTRL  PHONE NUMBER LIST    LOCAL ID SEQUENCE
    LEN                           REMOTE ID SEQUENCE
```

The headings denote the following information:

HEADING	MEANING
LDN	Logical device number.
PM	Port Mask (only valid for HSI)
PRT	Protocol.
LCL MOD	Local mode.
TC	Transmission code.
RCV TMOUT	Receive timeout.
LCL TMOUT	Local timeout.
CON TMOUT	Connect timeout.
MODE	O = Dial out. I = Manual answer. A = Automatic answer. D = Dual speed. H = Half speed. C = Speed changeable.
TRANSMIT SPEED	Transmission speed. (characters per second).
TM	Transmission mode.
BUFFER SIZE	Default buffer capacity, in words.
DC	Driver changeable or not changeable.
DRIVER OPTION	Driver options.
DEVICE CLASSES	Class name assigned to device
LDN	Logical device number
PHONE NUMBER LIST	A single telephone number — the default for the communications line.
LOCAL ID SEQUENCE	The default identification of the local computer.
REMOTE ID SEQUENCE	The default identification of the remote computer.

To suppress this listing, enter NO.

Step No.	Dialog
7	HIGHEST DRT=<XX>.?

XX is a value denoting the present highest DRT entry number that can be assigned to a device.

To change XX, enter the new value desired. If the highest-numbered device in the configuration is a device that uses more than one DRT entry (such as a terminal controller with one or two data set controllers), be sure to enter the *highest* of the DRT numbers.

To maintain the current XX, enter a carriage return.

8 LOGICAL DEVICE #?

To specify a device to be added or removed, enter the logical device number of that device.

To skip to Step 51 enter zero or a carriage return.

9 DRT # ?

To add a device, enter its DRT entry number. In the case of all data communications devices, the DRT number you assign must be the logical device number of the associated SSLC, INP, or HSI preceded by a pound sign (#). For example: #20.

To delete a device, enter 0 as the DRT entry number. The dialog will skip back to step 8.

NOTE: ADCC-connected terminals (on Series 30/33/40/44) are handled like single unit SID controllers. Thus, each has its own DRT. Each ADCC handles up to 8 terminals. If port 0 starts at DRT 8_{10} (%10) then port 2 on that same ADCC would be DRT 10_{10} (%12) and port 7 would be DRT 15_{10} (%17). The system console must be configured as DRT 8, the lowest configurable DRT on Series 30/33/40/44 systems.

10 UNIT #?

Enter the physical hardware unit number of the device, if the device shares its controller with other devices.

NOTE: All ADCC-connected terminals must be configured as UNIT 0.

If you are configuring a data communications subsystem, refer to the tables in Appendix B for unit numbers. Note that for DSN/MTS multipoint terminals, unit numbers identify the terminal by group ID and device ID. The numbering scheme is explained in the DSN/MTS Reference Manual. For

Step No.	Dialog
	DSN/MRJE, you must specify unit numbers for pseudo devices as shown in Table B-22. If the device does not share its controller with other devices, enter zero to continue.
11	SOFTWARE CHANNEL #? If the device is to be on a multicontroller channel, enter channel number; if not, enter zero. Enter zero for all data communications devices and pseudo devices.
12	TYPE? Enter the device type. Use one of the decimal numbers listed below. NOTE: The following device types are not supported on Series 30/33/40/44/64 systems: 1, 9, 18-20, 23, 33-37. 0 = Moving-Head Disc 1 = Fixed-Head Disc 2 = Flexible Disc 3 = 7911, 7912 or 7933 Disc, or Integrated Cartridge Tape Unit. 8 = Card Reader 9 = Paper Tape Reader 16 = Terminals, DSN/DS pseudo terminals, multipoint terminals, multipoint supervisor, asynchronous terminal controller. 17 = Intelligent Network Processor (DSN/INP) 18 = Synchronous Single-Line Controller 19 = Hardwired Serial Interface 20 = Printing Reader/Punch 22 = DSN/MRJE, Pseudo Device 23 = Programmable Controller 24 = Magnetic Tape 32 = Line Printer 33 = Card Punch 34 = Paper Tape Punch 35 = Plotter, CalComp Series 500 36 = Plotter, CalComp Series 600 37 = Plotter, CalComp Series 700 41 = DSN/DS Communications Driver
13	SUB TYPE? Enter the device subtype in the range 0 to 15.

Step No. Dialog

Refer to Appendix A for a summary of subtypes for all devices and subsystems.

NOTE

If you are configuring a terminal (type 16 or 32), the dialogue continues to step 14. If you are configuring an HSI (type 19), the dialogue skips to step 16. If you are configuring an SSLC (type 18), the dialogue skips to step 17. If you are configuring an INP (type 17), the dialogue skips to step 20. For all other device types, the dialogue skips to step 34.

14 TERM TYPE?

This question is asked only if device type is 16 or 32 (subtypes 14 and 15). To specify a default terminal type to be used at log on, enter one of the following numbers:

0 HP 30124A (HP 2749B), ASR-33
1 ASR-37
2 ASR-35
3 Execuport 300
4 HP 30123 (HP 2600A), or Datapoint 3300
5 Memorex 1240
6 HP 30120A (HP 2762A/B), GE Terminet 300 or 1200
7 (Reserved)
8 (Reserved)
9 HP 30122A (HP 2615A) BeeHive Mini Bee
10 HP 2621A, HP 2640A/B, HP 2641A, HP 2644A, HP 2645A, HP2647A, HP 2648A. Character mode or control of block mode transmission by a user program.
11 HP 2640A/B, HP 2641A, HP 2644A, HP 2645A, HP 2647A, HP 2648A. Allows use of block mode without program control of block mode transmission. May not be used for block/line mode.
HP 2645K Katakana/Roman Data Terminal
12 HP 2645K Katakana/Roman Data Terminal
13 HP 2601A. Also, for use when connected to a message switching network or other computer. Configured with ECHO disabled and no delay after carriage control operations.
14 Multipoint Terminal and Multipoint Supervisor.
15 HP 2635A Printing Terminal. 8-bit protocol (for second character set).
16 HP 2635A Printing Terminal. 7-bit protocol (standard character set).
18 Simple Terminal. No ENQ/ACK handshake, no prompt, no DC1 character and no delay character.
19 HP 2631B Remote Printer.

Step No.	Dialog

14 (Cont.)

The different systems offered by Hewlett-Packard support the following term types:

	TERM TYPE																	
System	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	19
II/III	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
30/33					x		x			x	x	x	x	x		x	x	x
40/44					x		x			x	x		x	x		x	x	x
64						x	x			x	x		x	x		x	x	x

NOTE

For DSN/DS pseudo terminals, or if the terminal type is not known, enter a carriage return.

15 SPEED IN CHARACTERS PER SECOND?

This question is asked only if device type is 16. Specify zero or a carriage return for all DSN/DS pseudo terminals, the multipoint supervisor, and DSN/MTS multipoint terminals. For other terminals, to specify the terminal speed in characters per second, enter 10, 14, 15, 30, 60, 120, 240, 480, or 960. The Series 30/33/40/44/64 also supports 480 and 960. (Required for subtypes 4-7, not used for subtypes 0-3 except when allocated programmatically.)

Otherwise, enter 0 or a carriage return to cause speed sensing at log on (subtypes 0-3 only).

NOTE

The dialog skips to Step 34.

16 PORTMASK?

This question is asked only if device type is 19 (Hardwired Serial Interface (HSI)). The values allowed are shown below and must be entered in decimal. This forms a mask indicating which HSI channel (port) will be used. Only one of the four channels may be designated for each unique logical device number.

8 = HSI cable connector CHAN 0
4 = HSI cable connector CHAN 1
2 = HSI cable connector CHAN 2
1 = HSI cable connector CHAN 3

17 PROTOCOL?

Enter 1 for Binary Synchronous Communication.

Step No.	Dialog
18	LOCAL MODE? 1 = HSI 1 or 2 = SSLC (See table B-14) 1 = Local is a primary contention station 2 = Local is a secondary contention station To resolve the contention problem in point-to -point operations, each station is assigned a priority (primary or secondary). Because the secondary station can gain control of the line for a transmission only when the line is left free by the primary station, the SSLC is usually configured as a primary station. Local mode determines the amount of time a local station will wait in response to a line bid; primary station timeout is two seconds and secondary is three seconds. If a response from the remote system is not received within the allowed time (two seconds for primary or three seconds for secondary), the line bid is re-transmitted until the number of retries permitted by the communications software is exhausted.
19	TRANSMISSION CODE? Enter the appropriate number for the transmission code in use. The code numbers are: 1 = Automatic code sensing of ASCII and EBCDIC if initially receiving ASCII if initially sending; or for Hardwired Serial Interface. 2 = ASCII 3 = EBCDIC Select the most frequently used method of transmission because your response establishes the configuration default. Users not transmitting by the default method can override this specification.
20	RECEIVE TIMEOUT? A value from 0 to 32,000. Default is 20 seconds. Entering 0 disables the timeout feature. Enter the positive number of seconds the communications software will wait to receive text before terminating the read mode. A batch job terminates if a timeout occurs. In a session, Receive Timeout is disabled. NOTE For all timeout responses (steps 20, 21 and 22), the subsystem displays an error when the communications software (CS) disconnects because of a timeout.

Step No.	Dialog
21	LOCAL TIMEOUT?

A value from 0 to 32,000. Default is 60 seconds. Entering 0 disables the timeout feature. Enter the positive number of seconds a connected local station will wait to transmit or receive before disconnecting.

The local timeout feature notes the time between requests to the communications interface.

The local timeout feature is disabled in a session.

22 CONNECT TIMEOUT?

A value from 0 to 32,000. Default is 900 seconds. Entering 0 disables the timeout feature. Enter the positive number of seconds the local station will wait after one attempt to make a connection to a remote station.

A connect timeout occurs if the "data terminal ready" signal fails to indicate line connection within the specified time.

NOTE

Steps 23 through 25 apply only to CS devices with switched lines connected through a modem (dial telephones, subtype 0). For CS devices with nonswitched lines connected through a modem (private lines, subtype 1), the dialogue skips to step 26. If the CS device is hardwired (subtypes 3 and 7), the dialogue skips to step 28.

23 DIAL FACILITY?

Enter YES when calls can be dialed from the local station. Enter NO when they cannot.

24 ANSWER FACILITY?

Enter YES if the local modem can answer calls, either manually or automatically. Enter NO if it cannot. A NO response causes the next step to be skipped.

25 AUTOMATIC ANSWER?

Enter YES if the local modem can automatically answer calls. Enter NO if manual answering is required.

Step No.	Dialog
26	DUAL SPEED? Enter YES if the local modem is dual speed (European modem). Enter NO if it is single speed. A NO response causes the next step to be skipped.
27	HALF SPEED? (This prompt relevant to European modem only.) Enter YES if the local modem is to operate at half speed. Enter NO if it is to operate at full speed. The dialog skips to step 29.
28	SPEED CHANGEABLE? Enter YES if the speed of the line is changeable. Enter NO if the line speed is fixed. You must respond YES if the line is hardwired and users will specify the transmission speed (that is, override the configured transmission speed).
29	TRANSMISSION SPEED? Enter the transmission speed of the line in characters per second. For the Hardwired Serial Interface, enter 250,000 for cable lengths less than 1000 feet and enter 125,000 for cable lengths greater than 1000 feet. For the Synchronous Single-Line Controller, enter the speed in characters per second (250, 300, 600, or 1200). For the Intelligent Network Processor, enter the speed in characters per second (250, 300, 600, 1200, 2400, 3600, 4800, or 7000). The transmission speed you specify is ignored for modems that provide internal clocking signals. This allows modems of different speeds to be used without reconfiguring the Operating System. The speed you specify becomes the system default. For DSN/DS and DSN/MTS, the Console Operator can override the default by including the *speed* parameter in the :DSCONTROL console command.
30	TRANSMISSION MODE? Enter the appropriate number for the transmission mode in use. 0 = Full duplex (i.e. constant carrier) 1 = Half duplex (i.e. switched carrier) SSLC AND INP Configure an SSLC or INP for Full Duplex (0) if your facility uses one of the following: • A leased line with four wire point-to-point installation. • A dial network with two lines (four wire equivalent). • A dial network with Wide Band Service. • A hardwired INP to INP.

Step No.	Dialog
30 (Cont.)	Configure an SSLC or DSN/INP for Half Duplex (1) if your facility uses one of the following: • A leased line four wire multidrop installation. • A dial network with a single-line (two wire) installation. • An HP Data Link • A hardwired SSLC to DSN/INP. Your response must agree with the remote system's configuration and with the characteristics of the communication line. HSI Always configure an HSI as Full Duplex (0).
31	PREFERRED BUFFER SIZE? Enter the desired buffer in words, up to a maximum of 4095 for an HSI or SSLC, or 1024 for an INP. Large buffer sizes can increase transmission efficiency depending on the error rate of the line, but use up memory space. Match buffer sizes whenever possible, because the effective buffer size that can be used is the smaller of the two buffer sizes between sender and receiver. For DSN/MTS subsystems, your response must correspond to the strapped buffer size on the Multipoint Communications Printed Circuit Assemblies in the multipoint terminals.
32	DRIVER CHANGEABLE? No for HSI and DSN/INP No for SSLC for DSN/DS and DSN/RJE. Yes for SSLC for DSN/MRJE and DSN/MTS.
33	DRIVER OPTIONS? Enter 0. NOTE The dialogue skips to step 43.
34	RECORD WIDTH? Enter the record width (in decimal words) for the device. Default widths are referenced in Appendix A. Disc device defaults should be used. However, for other devices, any record width up to the maximum may be specified for your configuration.
35	OUTPUT DEVICE? If the device is ever used as a job or session input device, enter the class name or logical device number to be used for the corresponding job/session listing device.

Step No.	Dialog
	There are advantages in using class names under certain circumstances. Suppose there are two line printers and two card readers in the system. Both line printers are in class LP. You may configure the output device for both card readers as LP. In this way, either card reader can acquire *either* one of the line printers dynamically (provided at least one line printer is unallocated) at run time. If this device is not a job/session input device, enter zero.
36	ACCEPT JOBS/SESSIONS? To specify that this device can accept a job or session input stream (J), enter YES Otherwise, enter NO. Note: Disc devices should *not* be job accepting.
37	ACCEPT DATA? To specify that this device can accept data external to a job or session input stream (A), enter YES. Otherwise, enter NO. Note: Disc devices should *not* be data accepting.
38	INTERACTIVE? To specify that this is an interactive device (I), enter YES. Otherwise, enter NO. Note: Disc devices should *not* be interactive.
39	DUPLICATIVE? To specify that this is a duplicative device (D), enter YES. Otherwise, enter NO. Note: Disc devices should *not* be duplicative.
40	ENABLE SEEKAHEAD? This question is only asked when a disc (Type=0 or 1) is being configured. If disc is configured as more than 1 unit, answer NO. Otherwise, enter YES to enable overlapped seeks if desired. Note: For Series II/III only.

Step No.	Dialog
41	INITIALLY SPOOLED? To designate this device as being spooled at cold-load (S), enter YES. Otherwise, enter NO.
42	INPUT OR OUTPUT? This question is asked only if the device is initially spooled and the device is an input/output device. Enter IN or OUT.
43	DRIVER NAME? Enter the name of the program file containing the driver for this device (Appendices A and B). For drivers written and supplied by the user, this name must contain from one to eight alphanumeric characters, beginning with a letter. (If the driver name is preceded by an asterisk, the driver will reside permanently in main-memory.) NOTE Steps 44 through 47 apply to CS devices with switched lines (device types 17 and 18, subtype 0). *The dialogue for all other devices skips to step 48.*
44	PHONE LIST? You can supply one number (usually a frequently dialed number) which will be the system default. Note that DSN/MRJE does not use this number. Instead, it uses the number configured in the DSN/MRJE configuration file. Enter YES to provide a default phone number. Enter NO if none is to be provided. A NO response causes the next step to be skipped.
45	PHONE NUMBER? Enter a string of not more than 30 characters.

Step No.	Dialog

46 LOCAL ID SEQUENCE?

Only DSN/DS recognizes local ID sequences.

The default local ID sequence can be specified in terms of code or number system. Enter a carriage return for a null local ID sequence. Enter one of the letters below, followed by the ID sequence in quotes, if code, or parentheses, if number system:

A	= ASCII	Example:	A "JOE"
E	= EBCDIC	Example:	E "STRING"
O	= Octal	Example:	O (7, 35, 5)
H	= Hexadecimal	Example:	H (A1, 1F, BB)

NOTE

Do not enter more than 16 characters for the local or remote ID sequence.

47 REMOTE ID SEQUENCE?

Only DSN/DS recognizes remote ID sequences.

Enter the default remote ID sequence in the same format as the local ID sequence (above). This can be repeated until a carriage return is entered.

48 DEVICE CLASSES?

Enter a list containing a device class name (up to eight alphanumeric characters, beginning with a letter). Class names are separated from each other by commas. These names are left to the discretion of the System Supervisor. They will be used in certain file commands or intrinsics when any member of a group of devices (such as any disc drive) can be referenced. *No name need be entered.*

A device can belong to more than one class, such as DISC and FHDISC. Only the classes DISC and SPOOL (if spooling is desired) are specifically required by MPE. DISC is the default device class for building files. SPOOL is the device class for designating "spooling discs." Spoolfiles will only be allocated on discs which are included in the special device class SPOOL. User files also may reside on spooling discs.

The following classes must also be included if the User Logging Facility will be used with devices in a class other than DISC:

TAPE	Must refer to devices in a magnetic tape class.
SDISC or CTAPE	Must refer to devices in a serial disc class.

If step 49 is not necessary, control returns to step 8.

Step No.	Dialog
49	IS classname A SERIAL DISC CLASS?

This question is asked (and it is asked for each class being implicitly created by this ldev change) if the device being configured is a supported serial disc device, and classname is not the name of an existing device class.

Enter YES or NO.

Note that the response to this question should be NO for all classes of system and private volume discs.

If the answer is YES, control returns to step 8.

50 IS Classname A FOREIGN DISC CLASS?

Answer YES or NO

Note that a "YES" response is appropriate only if the class is NOT a serial disc class.

This question will be asked only if question 49 was answered "NO". Control returns to step 8.

51 MAX # OF OPENED SPOOLFILES = < XXX > .?

To change the maximum number of input and output spoolfiles which can be FOPENed at one time, enter the new limit. To retain the current value, enter a carriage return. This can be used to control the generation of output spoolfiles.

The highest number you can specify as a maximum is dependent upon the number of logical devices on your system. The algorithm for determining this maximum number of spoolfiles is:

$$\frac{16244 - (\text{number of logical devices} \times 4)}{30}$$

NOTE

Each concurrent batch job to be executed requires at least two spoolfiles — one for standard input and one for standard output. Thus, if you specify a maximum of 20 spoolfiles in response to the above prompt, the system is restricted to running ten concurrent jobs.

52 LIST I/O DEVICES?

To print a listing of the new input/output device configuration, enter YES. This list appears in the format described in step 5.

To suppress the list, enter NO.

NOTE

The prompt in step 53 below, appears only if you have configured a CS device into the system.

Step No.	Dialog
53	LIST CS DEVICES? Enter YES to list the characteristics of the new CS device configuration. Enter NO to suppress the listing.
54	CLASS CHANGES? If you plan to add, delete, or change device class specifications, enter YES and continue to step 55. Otherwise, enter NO. If there are no errors in classes the dialogue skips to 66, else returns to step 4.
55	LIST CLASSES? To list the device classes and the logical devices contained therein, enter YES. To suppress the listing, enter NO. The format of the listing is: CLASS NAME — ACCESS TYPE — LOGICAL DEVICES Where: CLASS NAME shows the classes specified in the I/O configuration. ACCESS TYPE is in the form: IN - serial input (device types 8-30) OUT - serial output (device types 16-39) DA - direct access (device types 0-6) FD - foreign disc (device types 0, 2, 3) SD - serial disc (device types 0, 2, 3) I/O,C - input/output, concurrent devices (device types 16-23) I/O,NC - input/output, non-concurrent devices (device types 16-30) LOGICAL DEVICES are the logical device numbers of all devices specified for this class in the I/O configuration.
56	DELETE CLASSES? To delete previously defined classes, enter YES. Otherwise, enter NO to skip to step 58.
57	CLASSES? Enter names, separated by commas, of classes to be deleted.

Step No.	Dialog
58	ADD CLASSES? To define new classes or to add devices to previously defined classes, enter YES. To skip to step 65, enter NO.
59	CLASS NAME? To define a new class or to add devices to a previously defined class, enter class name. To skip to step 65, enter a carriage return.
60	LOGICAL DEVICE # 'S? Enter logical device numbers, separated by commas, in the preferred order of allocation. If it is not necessary to ask question 61 or 63, then a return is made to step 59.
61	IN, OUT, OR IN/OUT? Depending upon the types of the devices within the class, this question is asked to determine the desired device class access type. Enter: IN OUT IN/OUT If your response to this question was IN/OUT, then you are asked an additional question:
62	CONCURRENT OR NON-CONCURRENT? Are all the devices in the class capable of concurrent IN/OUT? Enter: NC C If it is not necessary to ask step 63, the dialogue returns to step 59.

Step No.	Dialog
63	SERIAL DISC CLASS?

This question is asked if all devices in the class are supported serial disc devices.

Enter YES if the classname is to be used exclusively for access of serial disc devices.

Enter NO for all other device classes.

NOTE

It is possible to configure a disc drive into two classes, one of which is a serial disc class and the other which is not, thus giving the drive the potential for use in either class.

The dialogue returns to step 59 if the answer is "YES".

64 FOREIGN DISC CLASS?

This question is asked ONLY if the class is NOT a serial disc class.

Enter YES if the classname is to be used exclusively for access of foreign discs.

Enter NO if the classname is not foreign disc.

65 LIST CLASSES?

To list the device classes and logical devices contained therein, enter YES.

If there are any errors in classes the dialogue returns to step 4.

66 LIST I/O DEVICES?

To print a listing of the new I/O configuration, enter YES.

To suppress the listing, enter NO.

NOTE

The prompts in steps 67 through 73, appear only if a CS device is configured or if additional drivers exist (for the CS driver-changeable option). If neither case exists, the dialogue skips to step 4.

Step No.	Dialog
67	ADDITIONAL DRIVER CHANGES?
	To prepare for additional driver changes, enter YES.
	If NO is entered, dialog skips to 4.
	NOTE The prompts in steps 68 through 70, below, appear only if additional drivers are already configured; *otherwise, the dialogue skips to step 71.*
68	LIST ADDITIONAL DRIVERS?
	To print a listing showing the presently-configured additional drivers, enter YES.
	To suppress this listing, enter NO.
69	DELETE DRIVERS?
	To delete an existing additional driver, enter YES.
	To skip to step 71, enter NO.
70	DRIVER NAME?
	Enter the name of the driver to be deleted. This prompt is repeated until you enter a carriage return, or until all drivers are deleted.
71	ADD DRIVERS?
	To prepare for adding drivers, enter YES.
	To skip to step 73, enter NO.
72	DRIVER NAME?
	Enter the name of the driver to be added. This prompt is repeated until you enter a carriage return, or until the maximum of 32 drivers have been added.
73	LIST ADDITIONAL DRIVERS?
	To print a listing showing the presently-configured additional drivers, enter YES.
	To suppress this listing, enter NO.
	GO TO STEP 4.

Step No.	Dialog
74	SYSTEM TABLE CHANGES? To prepare for changing the CST, DST, PCB, IOQ, or MTAB, or other parameters relating to memory usage, enter YES. To bypass these changes, and proceed to step 92, enter NO.
75	CST = <XXX>.? To change the size of the shareable portion of the CST from XXX entries to another value, enter the new value. To retain the current value, enter a carriage return.
76	EXTENDED CST = <XXXX>.? To change the size of the program portion of the CST from XXXX entries to another value, enter the new value. To retain the current value, enter a carriage return.
77	DST = <XXXX>.? To change the size of the DST from XXXX entries to another value, enter the new value. To retain the current value, enter a carriage return.
78	PCB = <XXX>.? To change the size of the PCB table from XXX entries to another value, enter the new value. To retain the current value, enter a carriage return.
79	I/O QUEUE = <XXX>.? To change the number of the input/output queue entries permitted from XXX entries to another value, enter the new value. To retain the current value, enter a carriage return.

Step No.	Dialog
80	DISC REQUEST TABLE = <XXX>? To change the size of the disc request table from XXX to another value, enter the new value. To retain the current value, enter a carriage return.
81	TERMINAL BUFFERS PER PORT = <XXX>.? To change the number of terminal buffers per port in the system from XXX, enter the new value. To retain the current value, enter a carriage return. NOTE: Terminal buffers are shared among all terminals. Entering a number for terminal buffers "per port" will not necessarily limit a given terminal to that number of buffers.
82	SYSTEM BUFFERS = <XXX>? To change the number of system buffers in the system, enter the new value. to retain the current value, enter a carriage return.
83	SWAP TABLE = <XXX>? To change the size of the swap table from XXX to another value, enter the new value. To retain the current value, enter a carriage return.
84	PRIMARY MESSAGE TABLE = <XXX>? To change the size of the primary message table from XXX to another value, enter the new value. To retain the current value, enter a carriage return. (You should have at least as many primary message table entries as system buffers.)
84.1	SECONDARY MESSAGE TABLE = <XXX>? To change the size of the secondary message table from XXX to another value, enter the new value. To retain the current value, enter a carriage return.
85	SPECIAL REQUEST TABLE = <XXX>? To change the size of the special request table from XXX to another value, enter the new value. To retain the current value, enter a carriage return.
86	ICS = <XXXX>? To change the number of words in the interrupt control stack (ICS), enter the new value. To retain the current value, enter a carriage return.
87	UCOP REQUEST QUEUE = <XXX>.? To change the number of entries allowed in the user controller process request queue to another value, enter the new value. To retain the current value, enter a carriage return.
88	TIMER REQUEST LIST = <XXX>.? To change the maximum number of concurrent time-out requests for the system clock allowed, enter the new value. To retain the current value, enter a carriage return.

Step No.	Dialog
89	BREAKPOINT TABLE = <XXX>.? To change the size of the breakpoint table from XXX entries, enter new value < 256. To retain the current value, enter a carriage return.
90	MAX NUMBER OF USER LOGGING PROCESSES = <XXX>? To change the maximum number of processes allowed on the system, enter the new value. To retain the current value, enter a carriage return.
91	MAX NUMBER OF USERS PER LOGGING PROCESS = <XXX>? To change the maximum number of users per process, enter the new value. To retain the current value, enter a carriage return.
92	MISC CONFIGURATION CHANGES? To prepare for the following miscellaneous configuration changes, enter YES: • Listing and (optionally) deleting global resource identification numbers (RIN's) assigned to users. • Number of RIN's available in the RIN pool. • Maximum number of global RIN's available. • Number of seconds allowed for logging-on. • Maximum number of jobs allowed on the system. • Maximum number of concurrent sessions allowed in execution. • Default central-processor time-limit for jobs. • Message catalog changes. To bypass these changes and proceed to step 108, enter NO.

Step No.	Dialog
93	LIST GLOBAL RINS?

To list the currently-assigned global resource identification numbers (RIN's), enter YES. To suppress this listing, enter NO.

The listing consists of the RIN number and the name of the user and account to which it is assigned (for each RIN).

94	DELETE GLOBAL RIN? (RELOAD option only.)

To prepare for deleting any of the currently-assigned global RIN's, enter YES. To bypass deletion and skip to step 97, enter NO.

95	ENTER RIN NUMBER?

To delete a currently-assigned global RIN, enter the RIN number.

This step is repeated until a carriage return is entered.

NOTE

Since global RIN's are permanently assigned to users and the RIN numbers will be hard-coded into their programs, RIN's should be deleted with caution.

For this same reason the most up-to-date RIN table (which resides on disc) is used when the system is cold-loaded, except in the case of a RELOAD. This implies that any changes to the RIN table occurring during a :SYSDUMP operation, including changes to the size of the table, only take effect when the tape produced by :SYSDUMP is cold-loaded using the RELOAD option.

Step No.	Dialog
96	LIST GLOBAL RINS?
	To list the updated global RIN's (as in step 93), enter YES.
	To suppress the listing, enter NO.
97	# OF RINS MIN = <YYY>, MAX = (XXXX).?
	To change the number of RIN's available in the RIN pool, enter a new value for XXXX. This value must be at least as great as YYY. (YYY is the maximum of 5 and the highest currently-assigned global RIN number.)
	To maintain the current maximum, enter a carriage return.
98	# OF GLOBAL RINS USED = <YYY>,MAX = <XXXX>.?
	To change the maximum number of global RIN's available, enter a new value for XXXX. Because of the current assignment of global RIN numbers, this must be at least as great as YYY.
	To maintain the current value, enter a carriage return.
99	# OF SECONDS TO LOG ON = <XXX>.?
	To change the number of seconds allowed for logging-on, enter the new value.
	To retain the current value, enter a carriage return.
100	MAX # OF CONCURRENT RUNNING SESSIONS = <XXXX>.?
	To change the maximum number of sessions allowed in execution at one time, enter the new value.
	To retain the current value, enter a carriage return.
101	MAXIMUM # OF CONCURRENT RUNNING JOBS = <XXX>.?
	To change the maximum number of jobs allowed in execution at one time, enter the new value.
	To retain the current value, enter a carriage return.
102	DEFAULT JOB CPU TIME LIMIT = <XXXXX>.?
	To change the value, enter the new value in seconds. A zero implies that jobs are not limited; sessions are limited only if the user supplies a limit on the :HELLO command.
	To retain the current value, enter a carriage return.

Step No.	Dialog
103	MESSAGE CATALOG CHANGES? To create a new message catalog from supplied file, enter YES. To skip to step 105, enter NO or a carriage return.
104	CATALOG INPUT FILE NAME? Name of disc file or formal designator to define the input file from which new catalog is to be built.
105	(Series 30/33/40/44/64) SOFTDUMP COMMAND CHANGES? To make changes, enter YES. For no changes, enter NO. Skips to step 108.
107	(Series 30/33/40/44/64) SOFTDUMP COMMAND FILE NAME? Enter file name. Dialog will respond with: NEW COMMAND FILE CREATED
108	LOGGING CHANGES? To prepare for changes to the logging characteristics of the system, enter YES. To bypass such changes and proceed to step 115, enter NO.
109	LIST LOGGING STATUS? To print a list of the events that can be logged and whether or not they are currently being logged, enter YES. To suppress the listing, enter NO.
110	STATUS CHANGES? To prepare for changes to the logging status, enter YES. If no changes are desired, enter NO to skip to step 113.

Step No.	Dialog
112	LIST LOGGING STATUS? To list the updated logging status, respond with YES. To suppress the listing, enter NO.
113	LOG FILE RECORD SIZE (SECTORS) = <XX>.? To change the value of the log file physical record size, enter the number of sectors desired. This number determines the size of the buffer for entries in the log file. (A sector is equal to 128 words.) To retain the current value, respond with a carriage return.

Step No.	Dialog

111 ENTER TYPE, ON/OFF?

You should enter the type number of the event (defined below), a comma, and ON to signify that it is to be logged or OFF to signify that it is not. The following Events may be logged:

Type No.	Event
1	Logging enabled
2	Job initiation
3	Job termination
4	Process termination
5	File close
6	System shutdown
7	Power failure
8	Spooler's spoolfile done record
9	Line disconnection
10	Line close
11	I/O error
12	Physical mount/dismount log record
13	Logical mount/dismount log record
14	Tape labels log record
15	Console log record
16	Program file event
17	Call process signals
18	DCE provided information
46	MPE maintenance request log record
47	Diagnostic control unit record

NOTE

Event 1 must be ON for any logging to take place. Turning event 1 OFF will disable all types except 11, 46 and 47. If event 2 is on and logging is enabled, the default input priority for jobs and sessions is 8; if event 8 is on and logging is enabled, the default output priority is 8. Otherwise, the normal default is 13 for both input and output priorities. Events 11, 46 and 47 will always be enabled. Requests to disable these types will be ignored. Step 111 is repeated until a carriage return is entered.

Step No.	Dialog
114	LOG FILE SIZE (RECORDS) = <XXXXX>.?

To change the maximum number of physical records permitted in the log file, enter a new value. The log file has 16 extents, so each extent will contain:

$$\left(\frac{\text{(log file size)}}{16} \times \text{(log file record size)} \right) \text{ sectors of disc space.}$$

To retain the present value, enter a carriage return.

115 DISC ALLOCATION CHANGES?

To prepare for disc allocation changes, enter YES.

To bypass such changes and proceed to step 129, enter NO.

116 DIRECTORY USED =<YYYY>,MIN = <ZZZZ>,MAX = <XXXX>.?

To change the maximum size of the directory from XXXX sectors, enter the new value; YYYY specifies the amount of directory currently used; ZZZZ specifies the minimum value to which XXXX can be set. (ZZZZ will often be greater than YYYY due to unused areas that are not at the end of the space allotted to the directory.) Maximum size cannot exceed 65,000 sectors.

To retain the present maximum size, enter a carriage return.

117 LIST VOLUME TABLE?

To list the disc volumes and their currently-assigned logical device numbers, enter YES. The listing is printed in the following format:

VOLUME #	NAME	LOG DEV #
volnumber	*volname*	*ldn*
.	.	.
.	.	.
.	.	.

In this listing, *volname* is a name of up to eight alphanumeric characters, beginning with a letter, identifying the volume; *ldn* is the logical device number assigned to that volume.

To suppress this listing, enter NO.

Note: Volume table changes take effect during RELOAD only.

Step No.	Dialog

118 DELETE VOLUME?

To prepare to delete a volume, enter YES. To bypass deletion and skip to step 120, enter NO.

119 ENTER VOLUME NAME?

To delete a volume, enter the volume name. (When the name is entered, the question is repeated.) Otherwise, enter a carriage return.

120 ADD VOLUME?

To prepare to add a volume, enter YES. To bypass addition and skip to step 122, enter NO.

121 ENTER VOLUME NAME

To add a volume, enter the volume name. (When the name is entered, the question is repeated.) Otherwise, enter a carriage return.

122 LIST VOLUME TABLE?

To list the disc volumes and their currently assigned logical device numbers (as in step 117), enter YES. In this listing, volumes just added (in step 120) will have logical device numbers of zero. To suppress this listing, enter NO.

123 VIRTUAL MEMORY CHANGES?

To change the allocation of disc space used for virtual memory, enter YES. To retain the current allocation, enter a carriage return. Dialogue skips to step 127.

Virtual memory allocation for LDEV1 can be changed only during a RELOAD. Allocation for other discs can also be changed during COOLSTART, COLDSTART or UPDATE.

124 LIST VIRTUAL MEMORY ALLOCATION?

To list the disc volume, logical device number, and virtual memory allocation, enter a YES. To suppress listing, enter a carriage return.

VOLUME NAME	LDEV#	VM ALLOCATION
MH7920U0	1	10

125 ENTER VOLUME, SIZE IN KILOSECTORS? (Can enter volume name or LDEV, size)

CAUTION

You must configure at least 1 kilosector of virtual memory on LDEV1.

Step No.	Dialog
126	VIRTUAL MEMORY CHANGES? Enter a YES, to loop back to step 124. Enter a carriage return to continue dialog.
127	MAX # OF SPOOLFILES KILOSECTORS = <XXXXXX>? To change the maximum number of sectors which can be allocated to spoolfiles (expressed in thousands of sectors), enter the new value. To retain the current limit, enter a carriage return.
128	# OF SECTORS PER SPOOLFILE EXTENT = <XXXXX>.? Enter the size, in sectors, for each spoolfile extent. This must be a value between 128 and 32767.
129	SCHEDULING CHANGES? This heading preserved to eliminate the need to alter batch job headings. Subheading reserved for future user. NOTE Scheduling changes are accomplished only by use of the :TUNE command.
130	SEGMENT LIMIT CHANGES? To prepare for changing the limits on code and data segments, enter YES. To retain the current limits and skip to step 137, enter NO.
130A	MAX # OF CONCURRENT RUNNING PROGRAMS = <XXX> ? To change the maximum number of concurrently running programs from <XXX>, enter the new value. To retain the current value, enter a carriage return.
131	MAX CODE SEG SIZE = <XXXXX>.? To change the maximum number of words allowed in any code segment from XXXXX, enter the new value. To retain the current value, enter a carriage return.

Step No.	Dialog
132	MAX # OF CODE SEGMENTS/PROCESS = <XX>.?
	To change the maximum number of code segments allowed any user process, enter the new value.
	To retain the current value, enter NO.
133	MAX STACK SIZE = <XXXXX>.?
	To change the maximum number of words allowed in any user stack from XXXXX, enter the new value. (A maximum value of 31232 is permitted.)
	To retain the current value, enter a carriage return.
134	MAX EXTRA DATA SEG SIZE = <XXXXX>.?
	To change the maximum number of words allowed in any extra data segment from XXXXX, enter the new value.
	To retain the current value, enter a carriage return.
135	MAX # OF EXTRA DATA SEGMENTS/PROCESS = <XXX>.?
	To change the maximum number of extra data segments that a process can have, enter the new value.
	To retain the current value, enter a carriage return.
136	STD STACK SIZE = <XXXX>.?
	To change the number of words initially assigned for a user stack (Z-Q area) by default (when the user specifies no value) at preparation time from XXXX, enter the new value.
	To retain the current value, enter a carriage return.
137	SYSTEM PROGRAM CHANGES?
	To prepare to replace a program belonging to the system, enter YES.
	To proceed directly to step 139, enter NO.

Step No.	Dialog
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?
	To replace a program belonging to the system, enter the name of the program, a delimiting comma, and the name of the program file which is to replace the program. The replacement program need not be in the public group of the system account, PUB.SYS. However, a fully qualified program file name will be required if the program file is not in the logon account/group structure.
	The question is repeated until a carriage return is entered.
139	SYSTEM SL CHANGES?
	To prepare for changes to the System Segmented Library (SL.PUB.SYS) enter YES. Otherwise, enter NO to skip to step 148.
140	LIST LIBRARY?
	To list the names of the code segments in the System Segmented Library and their entry-points and external procedures, enter YES.
	To suppress this listing, enter NO.
141	DELETE SEGMENT?
	To prepare for deleting a code segment from the System Segmented Library (SL.PUB.SYS), enter YES.
	To proceed directly to step 143, enter NO.
142	ENTER SEGMENT NAME?
	To delete a code segment from the System Segmented Library, enter the name of that segment. (When the segment name is entered, the question is repeated.)
	Otherwise, enter a carriage return.
143	REPLACE SEGMENT?
	To prepare for replacing a code segment in the System Segmented Library, enter YES.
	To proceed directly to step 145, enter NO.

Step No.	Dialog

144 ENTER SEGMENT NAME,USLFILE NAME [,S/C/P]?

To replace a code segment in the System Segmented Library, enter the name of the segment; a delimiting comma; and the name of the USL file where the replacement segment can be found. Also, optionally, enter a delimiting comma followed by one of these three characters:

None	Segment is placed in the library as a user segment, but is not permanently allocated.
S	To declare the segment to be a permanently-allocated system intrinsic segment (in virtual memory).
C	To declare the segment to be a main-memory resident system intrinsic segment.
P	To declare the segment to be a permanently-allocated user segment. (This option requests the same function as the :ALLOCATE command, defined in Section II.)

The question then is repeated.

Otherwise, enter a carriage return.

NOTE

If you enter a USLFILE name which is in error (typographic input error), an error message results and the Configurator proceeds back to step 141. In this regard the segment which was to have been replaced has, in fact, been *deleted*. You may proceed to step 145 and attempt to add the USL file.

145 ADD SEGMENT?

To prepare for adding a code setment to the System Segmented Library, enter YES.

Otherwise, enter NO to skip to step 147.

Step No.	Dialog
146	ENTER SEGMENT NAME, USLFILE NAME [,S/C/P]?

To add a code segment to the System Segmented Library, enter the name of the segment; a delimiting comma; the name of the USL file where the segment can be found. Also, optionally, enter a delimiting comma followed by one of these three characters:

None	Segment is placed in the library as a user segment, but is not permanently allocated.
S	To declare the segment to be a permanently-allocated system intrinsic segment (in virtual memory).
C	To declare the segment to be a main-memory resident system intrinsic segment.
P	To declare the segment to be a permanently-allocated user segment (in virtual memory). (This option requests the same function as the :ALLOCATE command, defined in Section II.)

The question then is repeated.

Otherwise, enter a carriage return.

NOTE

If you enter a USL file name which is in error (any error), an error message results and the Configurator proceeds back to step 141.

147 LIST LIBRARY?

To list the updated System Segmented Library, enter YES.

To suppress this listing, enter NO.

148 ENTER DUMP DATE?

To copy only the modified operating system to tape, enter a carriage return; the dialogue skips to step 151.

To copy the MPE system, the current accounting structure, and all files to tape (or serial disc), enter 0. This tape can then be used to RELOAD the system (Console Operator function). The tape can also be used with the :RESTORE command (described in Section II) to retrieve a file.

148 (Cont.) To copy the MPE system, the current account structure, and any files that were changed *on* or *after* a particular date, enter that date in the format *mm/dd/yy*. (In this format, *mm*, *dd*, and *yy* are one or two decimal digits representing the month, day, and year, respectively.) This tape can be used in conjunction with other tapes to RELOAD the system, or to retrieve one or more files by using the :RESTORE command.

NOTE

Because files in use with write, append, update or read/write access will not be copied, system back-up should be performed only when no users are logged onto the system.

149 ENTER DUMP FILE SUBSET(S)

At this point, you can instruct the Configurator to dump only certain files by entering file subsets. These subsets are entered in the fileset form as discussed under the :STORE command fileset parameter in Section II. In addition, you can use any of the options under the :STORE command (SHOW, FILES, DATE, and ONERROR). The *storefile* parameter is not allowed in the SYSDUMP command.

For a complete SYSDUMP, enter a carriage return or @.@.@. This is the case where the Configurator copies the system, the account, group, user, and file structure as well as all files on the system. This type of SYSDUMP is used periodically to ensure integrity of the system and all files. This operation should be done at a convenient time, when there are no other users on the system. If there are other users on the system and they are accessing some files, those files being accessed will not be copied to the storage medium. This defeats the purpose of a full system and file backup. Because you are saving files along with the system, the Configurator proceeds to step 150.

Filesets may be continued on a new line by entering an ampersand (&) as the last non-blank character of the present line; continue filesets on the next line. Be sure to enter the ampersand before you get to column 73. A *fileset* may not be divided between two lines. The total number of lines that may be entered is determined by the amount of available stack space remaining. If the total space is exceeded SYSDUMP will terminate with the message UNABLE TO OBTAIN STACK SPACE.

NOTE

SUBSET can be any *fileset* and/or option in the :STORE command format.

150 LIST FILES DUMPED?

To obtain a list showing the name of each file copied, enter YES. To suppress this list, enter NO. The optional listfile parameter to the :SYSDUMP command is useful to direct the list to a high speed printing device when it is known that a great many files exist on the system. A list showing the number (count) of files copied, the number of files not copied, the names of the files not copied, and the reasons why they were not copied is always provided. The formats for these listings are:

Step No.	Dialog

```
FILES STORED =

  FILE    .GROUP   .ACCOUNT   LDN    ADDRESS   VOLUME


FILES NOT STORED =

  FILE     .GROUP       .ACCOUNT  FILESET     REASON
```

151 The operator is now requested to assign the serial storage device on which you have arranged for a scratch tape, disc or cartridge tape to be mounted. After operator assignment, the system is copied to tape (multi-reel files) or serial disc. It then can be loaded and initialized as directed under the heading *System Start-Up and Modification*.

NOTE: If flexible discs are to be used, you must be certain that an adequate number of pre-serialized flexible discs are available (Series 30/33/40/44/64 only).

Check the list of files not dumped for user files you want to save. Frequently the files; LOADLIST, MEMLOG, SL, and LOGXXXX are open when SYSDUMP is running, so their names often appear on the list of files not dumped. By a special process, SYSDUMP records the system file, SL, on tape, so it is saved. (See Appendix F for the list of system files dumped.) The other three files mentioned above are recreated from scratch by the Initiator each time the system is started.

If any file belonging to the system is not copied, the message:

WARNING FOLLOWING SYSTEM FILES NOT DUMPED

is issued and the file name and the reason it was not copied is listed. If the file was to replace a system program, the program name follows in parentheses.

If a response other than a carriage return was entered in answer to the ENTER DUMP DATE? question in step 148, the list and count of files will be provided as described in step 150.

To denote termination of the Configurator/User Dialog, the following message is printed.

END OF SUBSYSTEM

EXAMPLE OF SYSTEM BACK-UP ONLY

The Configurator/user dialog proceeds with output from the Configurator (:SYSDUMP) and input from the System Supervisor user. In the following example, only the system will be saved on a serial storage device. Prior to entering the dialog, the user logs on to the system and inputs at least a file reference to a serial storage device. A typical example is:

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

(Assume that device class TAPE is a magnetic tape.) The dialogue between Configurator and user then commences as follows:

Step No.	Dialog
1	ANY CHANGES? NO Reply NO (or carriage return) to omit changing the system and to skip to step 148.
148	ENTER DUMP DATE? Enter carriage return. This is the null case of the Configurator. The operator is now requested to assign the serial storage device on which you have arranged for a scratch tape or disc to be mounted. After operator assignment, the system (only) is copied to tape or serial disc. In this regard, note that the account, group, and user structure is *not* copied to the device. However, the system programs in PUB.SYS are copied. NOTE: If flexible discs are to be used (Series 30/33/40/44/64), you must be certain that an adequate number of pre-serialized flexible discs are available.
151	To denote termination of the Configurator program, the following message is printed: END OF SUBSYSTEM

EXAMPLES OF SYSTEM BACK-UP WITH FILES

The Configurator/user dialog proceeds with the Configurator (:SYSDUMP) output and user input desired. In the first example below, the system and all files will be saved. In the second example, the system and a portion of the files will be saved. Prior to entering the dialog, the user logs onto the system and inputs at least a file reference to a serial storage device.

A typical example is:

```
:FILE SAVALL; DEV=TAPE
:FILE LISTIT; DEV=LP
:SYSDUMP *SAVALL, *LISTIT
```

Assume that device class TAPE and LP are a magnetic tape and a high speed printer respectively.

You may dump a maximum of 16,000 files with the :SYSDUMP command. To increase this maximum to 32,000 files, enter the :FILE command:

:FILE SYSDGOOD;DISC=32000,16,1

In this example to save the system and all files, the dialog is as follows:

Step No.	Dialog
1	ANY CHANGES? NO Enter NO (or carriage return) to omit modification to the system and to skip to step 148.
148	ENTER DUMP DATE? 0 Enter a zero.
149	ENTER DUMP FILE SUBSETS? At this point, you can instruct the Configurator to dump only certain files by entering file subsets. These subsets are entered in the fileset form as discussed under the :STORE command fileset parameter in Section II. For a complete SYSDUMP, enter a carriage return or @.@.@. This is the case where the Configurator copies the system, the account, group, user, and file structure as well as all files on the system. This type of :SYSDUMP is done periodically to ensure integrity of the system and all files. This operation should be done at a convenient time, when there are no other users on the system. If there are other users on the system and they are accessing some files, those files being accessed will not be copied to the storage medium. This defeats the purpose of a full system and file backup. Because we are saving files along with the system, the Configurator proceeds to step 150.
150	LIST FILES DUMPED? To obtain a listing showing the name of each file copied, enter YES. To suppress the list, enter NO. (A list showing the number (count) of files copied, the number of files not copied, the names of the files not copied, and the reasons why they were not copied is always provided.) The operator is now requested to assign the store device on which the :SYSDUMP will be placed.
151	To denote termination of the Configuration program, the following message is displayed: END OF SUBSYSTEM

In the next example, the Configurator is used to copy the system, all accounting structures, and some files. This particular use of the Configurator needs further explanation before proceeding with the example.

When the system is operational, any file which is modified in some way will have a modification data placed in the file label at the time of modification (during the file close process). Some examples are:

Writing on an existing file.
Creating a (new) permanent file.
Using the :RENAME command.

The process of reading a file or purging a file does not constitute a modification.

It can be quite time consuming to back up the system, all accounting structures, and all files every day. Thus an option is provided so that you periodically (for example, once a week) save everything, and then save various files that have been modified since the full system and file back up.

In the next example, assume that on Friday, November 18, 1977, you did a full system back up. On Monday, November 21, the users on the system created many new files. When there are no other users on the system, you, as the System Supervisor, log onto the system and enter a file reference to a store medium, a magnetic tape for example:

```
:FILE SAVESOME;DEV=TAPE
:FILE LISTSOME;DEV=LP
:SYSDUMP *SAVESOME,*LISTSOME
```

The dialog is as follows:

Step No.	Dialog
1	ANY CHANGES? NO Enter NO (or carriage return) to omit modifications to the system and to skip to Step 148.
148	ENTER DUMP DATE? 11/18/77 Enter the date on which the full system and file back up was made. The Configurator program (SYSDUMP) will save the system and all accounting structures. It will then compare each and every file label modification date to the one which was entered in step 148. Only those files whose modification date is the same or later than the date which you have input will be considered for saving. In this way, it can be seen that if the system had to be completely reloaded for any reason, it could be accomplished in two steps. First, you RELOAD the most recent tape (or serial disc), SAVESOME. When asked if there are more files, you mount the full system back up tape (or serial disc). This assures that the most recent versions of files are restored since RELOAD acts like RESTORE with the KEEP option.

Step No.	Dialog
149	ENTER DUMP FILE SUBSETS? At this point, you can instruct the Configurator to dump only certain files by entering file subsets. These subsets are entered in the fileset form as discussed under the :STORE command fileset parameter in Section II. To dump all files modified since 11/18/77, enter a carriage return or @.@.@.
150	LIST FILES DUMPED? To obtain a listing of all the files which have been dumped, enter YES. To suppress the listing, enter NO. A list showing the number (count) of files copied, the number of files not copied, the names of the files not copied, and the reasons why they were not copied is always provided.
151	To denote termination of the Configuration program, the following message is displayed. END OF SUBSYSTEM

KEEPING THE ACCOUNTING STRUCTURE OFFLINE

In the previous discussion of the Configurator dialogue under step 148, it was stated that your account, group, user, and file structure information is kept on the serial storage medium (except the option to save only the operating system). Experience has shown that it is a good idea to keep all of the created structures on some offline medium such as a magnetic tape or serial disc file or a job card deck. This requires extra effort on the part of the System Supervisor in order to keep the offline media current.

SYSTEM START UP AND MODIFICATION

Any person with access to the MPE system console can turn the system power on and off, start up MPE, alter the current input/output device configuration, and shut down the system. This person need not have System Manager or System Supervisor capability; in fact, because he does not need to log on to the system, he need not even have standard user capability. See the *MPE Console Operator's Guide* for power on/off procedures.

Start up of MPE, recovery of spoolfiles or the reconfiguration of the input/output devices and the system, and handling of disc errors are done through a program called the MPE Initiator (or INITIAL). The Initiator is a stand-alone program that creates an operational MPE using files and table information available to it from tape or, in certain cases, from disc. The Initiator program is brought into memory for execution via cold load.

The cold load operation loads a subset of the Initiator program into memory from disc or serial storage devices. The switches used to start the Initiator program vary with the II, III, 30, 33, 40, 44 and 64 systems. For Series II/III, press the RUN/HALT switch. For Series 30, 40, and 44, press the LOAD key on the System Control Panel; for Series 33, press the COLDLOAD key on the System Control Panel. (For both Series 30 and 33, on the System Console you can press the LOAD key while pressing the CNTL key.) For Series 64, type "LOAD" on the system console after engaging the Diagnostic Control Unit (DCU).

Once the Initiator program is started, it pauses until you press (RETURN) on the System Console, then begins an interactive dialog. Some flexibility is provided within the Initiator to change main memory size and input/output device configuration.

There are five types of cold load: WARMSTART, COOLSTART, UPDATE, COLDSTART, and RELOAD. (Refer to Table 6-1.) The subset of the Initiator is loaded into memory from the system disc during a WARMSTART or a COOLSTART. The subset is loaded from a backup serial storage medium during an UPDATE, a COLDSTART, or a RELOAD. The individual cold load options are described in the following paragraphs.

1. WARMSTART cold loads the system from the system disc. This is the procedure used if incompletely processed spooled jobs and spooled files must be recovered. On any other option, any spooled files found in the system are deleted.

   If a COOLSTART, COLDSTART, UPDATE or RELOAD does not complete for any reason, the next cold load may not be a WARMSTART. A violation of this results in an error message and halt.

NOTE

The Initiator program should never be halted. If error recovery is not possible, the Initiator program should be allowed to complete, i.e., run until the request for DATE? appears.

2. COOLSTART cold loads the system from the system disc. This is the standard operating procedure when a system is routinely shut down at night and brought up the next day. All permanent user files (including programs such as FORTRAN/3000, COBOL/3000, SPL/3000 and EDIT/3000 that run as MPE subsystems) are saved; but the operational enviornment present prior to the last shutdown is not retained. Thus, all jobs and sessions in progress at shutdown (before COOLSTART) are lost.

3. UPDATE cold loads the system from magnetic tape or from serial disc. The system files come from the backup medium while the I/O device configuration, directory, accounting information and global RINs come from the system disc (user's files remain undisturbed). This is the standard operating procedure when starting the system with an updated MPE tape from Hewlett-Packard or an MPE tape prepared for a different HP 3000 Computer System, and should be used ONLY in those situations. If an UPDATE does not complete, it must be followed by a COLDSTART, UPDATE, or RELOAD. Any violation of this procedure results in an error message and a halt.

4. COLDSTART cold loads the system from magnetic tape or serial disc. The system files and I/O device configuration come from the backup medium while the user files, directory, accounting information, and assigned resource identification numbers (global RIN's) are retained current on the system disc. This allows modification of the system configuration while retaining user information. COLDSTART is commonly used to allow an installation to keep several cold load tapes or discs, each with a different configuration. If a COLDSTART does not complete for any reason, it must be followed by a COLDSTART, UPDATE, or RELOAD. Any violation of this causes an error message and a halt.

5. RELOAD cold loads the entire system, including all system files and I/O configuration information, from the backup medium. This option assumes that there is no information on the system disc. If any user files are dumped on the backup medium, the directory, accounting information, assigned global RIN's and user files are restored to the system disc from the backup copy. If the accounting structure was not dumped, a directory is created with the SYS account, PUB group, and MANAGER user. RELOAD is normally the method used to install the first MPE system supplied by Hewlett-Packard. It also is used when restoring the system (from a tape or serial disc generated by the user through a :SYSDUMP command) following a disc crash. If a RELOAD does not complete for any reason, the next cold load must also be a RELOAD.

   When reloading from multiple volumes of tapes or serial discs that were created by the :SYSDUMP command, the first volume of the latest backup copy should be used for cold loading because it contains the up-to-date directory and accounting information. If not all of the files on the system are contained in this volume, an additional volume will be requested.

NOTE

Files belonging to non-system domain (private) volume sets are not restored during RELOAD.

No startup except an automatic power fail recovery permits resumption of user batch jobs or interactive sessions. Thus jobs and sessions must be re-initiated from the beginning by users after a system shutdown or a system failure.

All cold load options except WARMSTART allow you to alter the current I/O device configuration. This is done through an interactive dialog between the System Supervisor and the MPE Initiator.

Table 6-1 shows the origin of MPE system components for the various cold-load options.

When the system is started, the initialization of blank system disc packs, the writing of labels for them, and the renaming of volumes is accomplished by the MPE Initiator. Formatting of blank disc packs, however, is done by a diagnostic program rather than by the Initiator. See Appendix H.

Table 6-1. System Startup of HP 3000 Systems

types of cold load	subset of Initiation program loaded from	Primary use	allows alteration of I/O configuration	spool file recovery	permanent user files	information that comes from system disc	information that comes from backup medium (magnetic tape or serial disc)
WARMSTART	system disc	standard procedure if spool files and restartable jobs must be saved	no	yes	retained	all	none
COOLSTART	system disc	to bring up system after overnight shutdown	yes	no	retained	all	none
UPDATE	magnetic tape or serial disc	for updated MPE tape or MPE tape for different HP 3000 system	yes	no	retained	I/O and system configuration, directory, accounting information, global RINs	system files
COLDSTART	magnetic tape or serial disc	to keep several cold load tapes or discs with different configurations	yes	no	retained	directory, accounting information, global RINs	system files, I/O and system configuration
RELOAD	magnetic tape or serial disc	to install first MPE tape or after disc crash	yes	no	reloaded (if dumped on backup medium)	assumes no information on system disc	all (including users' files if dumped on backup medium

CHANGING I/O DEVICE CONFIGURATION

All cold-load options except WARMSTART allow the operator to alter the input/output device configuration currently in effect. This is done through an interactive dialogue between the user and the MPE Initiator. This dialogue is described later in this section.

Be aware however, that any changes you make in the I/O configuration during the Initiator dialogue will not be backed up on tape. Therefore, at your earliest convenience, use SYSDUMP to obtain a back-up for the changes you make.

Table 6-2. Source of MPE System Components

MPE COMPONENT	WARMSTART	COOLSTART	UPDATE	COLDSTART	RELOAD*
MPE Programs, System Library	disc	disc	serial storage device	serial storage device	serial storage device
I/O & System Configurations	disc	disc	disc	serial storage device	serial storage device
Accounting Info, File Directory, Volume Table & User Files	disc	disc	disc	disc	serial storage device
Spoolfiles & Jobs	disc	—	—	—	—

*Certain information which may reside on the disc is checked for validity. See discussion on DISC ORGANIZATION.

DISC ORGANIZATION AND ERROR RECOVERY

In order to appreciate how to recover disc errors, some background information is necessary so that you will have a better understanding of the error recovery facilities of the system.

The HP 3000 Computer System supports two distinctly different types of disc devices. The first is the *fixed-head* disc. These devices are normally used as the system disc because they are faster at accessing and transferring data than moving-head discs. They can do this because there is a magnetic "head" for each track on the disc. The head is fixed in position. The disc itself rotates past the heads and when data is transferred, the proper head is electrically switched on, so that data is transferred on the desired track. A *moving-head* disc has a magnetic head attached to an arm which is, in turn, attached to a servo mechanism. The arm must physically move to a specific cylinder before any data can be transferred. Moving-head discs are usually slower at accessing and transferring data than fixed-head discs. As an approximate comparison, fixed-head discs have an access time of approximately 8 to 11 milliseconds, while moving-head disc access time is in the 25-60 millisecond range.

Figure 6-1 is a block diagram showing how the operating system, directory and virtual memory are arranged on a typical system disc. The addresses change as configurations change, as Hewlett-Packard makes changes to the system, and as you change things (for example, the Directory or virtual memory).

For the purpose of this discussion, a typical user disc is one which contains a Volume Table, Defective Tracks Table (DTT) or Defective Sectors Table (DSCT), Disc Free Space Map and user files.

The operating system always addresses a disc by its logically ascending sector addresses. However, a physical disc is divided into cylinders, tracks, and sectors. Figure 6-2 shows a typical moving-head disc (the HP 7920) with the disc pack installed. Each inner platter of the disc pack has two surfaces used for storing information (except the servo platter which has one data surface). A cylinder is the area of all surfaces of a pack where all 5 heads can access a particular area of a platter surface. The head mechanism moves the entire group of heads from cylinder 0 through cylinder 822, depending on what cylinder has been requested by the software. A track is the area under one head on one surface. Therefore,

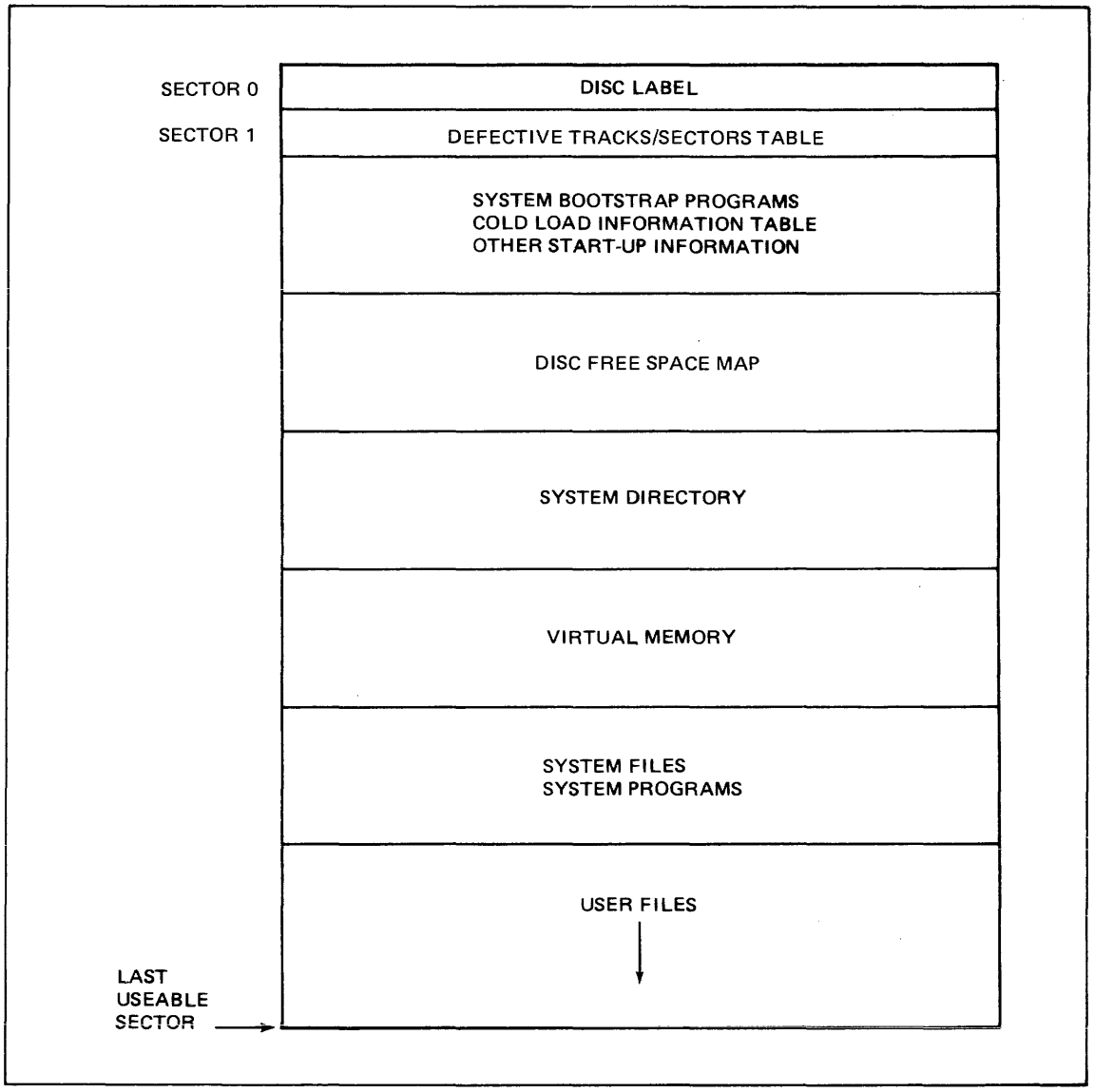


Figure 6-1. Typical System Disc Layout

in this case, we could access 5 tracks on cylinder 0 before being required to move the head mechanism to a new cylinder. Figure 6-3 shows the track positions on the left side of the surface and how the sectors are divided on a particular track on the right side of the figure. This type of disc head/cylinder, track/sector addressing is common for most moving-head discs. However, it should be viewed as an example only.

A disc pack, when it is newly manufactured, has no information on it. This is called an unformatted pack. In order to begin using the pack, it must contain information which the electronic hardware controllers need in order to determine where a track and sector are physically located on the disc. Additionally, parity checking and other information must be initially written on the disc at each track and sector. This process is called "formatting" the disc. For this system, formatting the disc is done by a stand-alone disc diagnostic. This diagnostic is normally executed by the Hewlett-Packard customer engineer. If you desire to format your own disc packs, please consult him. Also, see Appendix H for more information.

Once a pack is formatted, it can then be written with data. In MPE, it will be in the form of figure 6-1 or a user disc. In either case, *all* disc packs used by the system will have a *disc label* at track 0, sector 0 of the pack. The operating system refers to this name as a *volume* name. A *volume* name is an ASCII character string of up to eight alphanumeric characters beginning with a letter. *Volume* names are added to a disc pack when you are initiating the system using the initiator program, which is discussed later. The *volume* name is unique for each disc pack.

A table of volume names, known as the Volume Table, is kept on the system disc; it maps each volume name into the logical device number of the disc on which the volume is mounted. This allows disc packs of similar type to be mounted on any disc drive (for instance, switching packs among drives between cold-loads). For the UPDATE option, the initiator will ask for the DRT number of the system disc. This is required because the I/O configuration will not be read from the magnetic tape copy of the system. (The DRT entry of the system disc can be any entry.)

The System Directory tells the volume on which a particular file is to be found, and its sector address on that volume. A similar directory exists for spoolfiles. When the system is cold-loaded in any mode except RELOAD, the volume table that resides on the disc is used. The Initiator checks to ensure that all volumes defined previously are indeed mounted. This ensures that all files contained in the file directory are still there. A volume may be added but not deleted, since files defined in the directory may reside on that volume.

When the system is cold-loaded using the RELOAD option, the volume table on the cold-load tape is used. Volumes may be deleted or added, since each of the user files defined in the directory on the tape will be reloaded onto one of the available volumes.

NOTE

At the onset of a RELOAD, checks are made for the validity of the VOLUME label and the defective tracks/sectors table on disc. If the label and the table are not corrupt, then the Initiator will use the information on the disc, and will not rebuild a volume label or defective tracks/sectors table over the existing information.

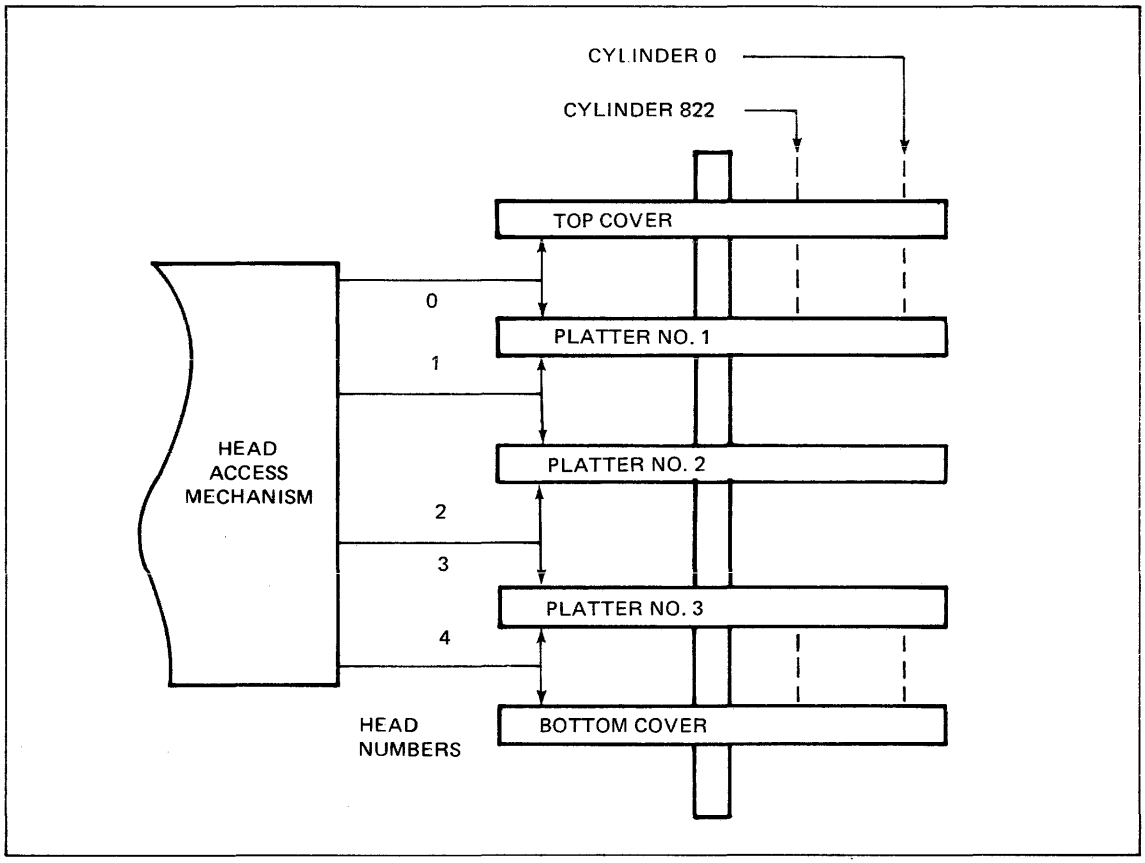


Figure 6-2. Disc Drive Structure

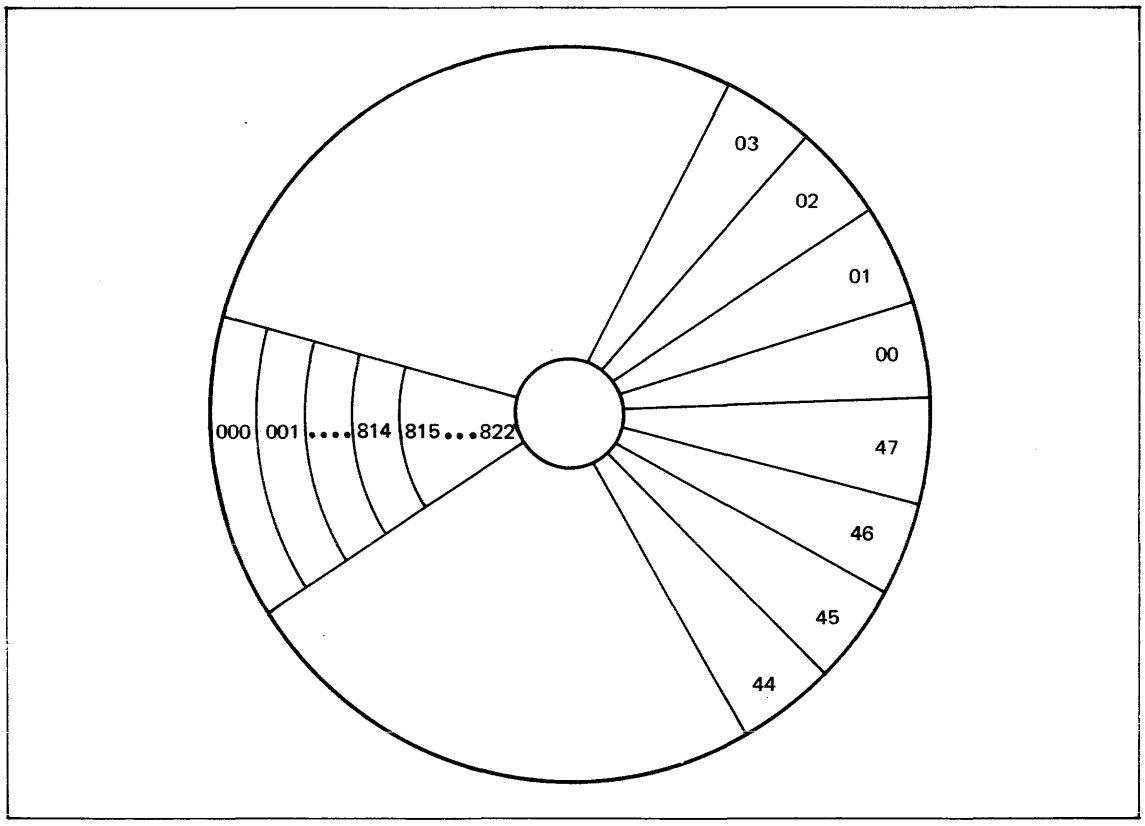


Figure 6-3. Disc Platter Format

The Initiator automatically will assign sector 1 of track 0 as the *defective tracks/sectors table* on a newly formatted pack.

The operating system software expects all system bootstraps, tables, files and programs to be in logically ascending order as generally shown in figure 6-1. Some means (transparent to the system software) of bypassing and/or reassigning of defective areas on moving-head disc packs is a necessity because disc packs cannot be made absolutely perfect. Fixed head discs, on the other hand, are manufactured much more precisely and rarely have defective areas on the recording surface. Due to the complexity of accessing a fixed-head disc, no means of reassigning defective areas is provided. However, a defective area can be bypassed (deleted).

There is no user dialog for handling defective areas of a type 3 disc. The MPE Initiator will automatically reassign any defective sector entries it finds in the Defective Sectors Table. The discussion below applies only to type 0 and type 1 discs.

When the MPE Initiator encounters a disc that does not have a valid MPE disc label, the System Supervisor is asked to give the disc a volume name. If the disc is a type 0 disc, the System Supervisor also is requested to define the logical pack size of the disc (in cylinders); this is the amount of space available for use by the system and user files. The remaining portion of the disc may be used to re-assign alternate tracks when a disc error occurs. For type 0 discs, a track-by-track read is then initiated. Any tracks found to be defective may be re-assigned on alternate tracks, or deleted (their space will not be available for use).

For example, let us suppose that a new cartridge disc pack is inserted into the system disc device. It is a type 0, subtype 2 device. The Initiator knows from its internal table that the minimum number of cylinders is 150 and the maximum number of cylinders is 203. At initialization time (a RELOAD operation), you are asked (by the Initiator) the following:

```
LOGICAL PACK SIZE (CYLINDERS) = 200?
```

You enter a carriage return to leave the assigned value. In this example, there would be three cylinders of tracks available to be used if some other area of disc had an error. Due to type and subtype of disc, there are four tracks per cylinder, making a total 12 tracks available to be used in case an error occurs.

Let us assume that the system is operational and a user is accessing (reading) a file which is located at tracks 700, 701, and 702 of some disc. Suppose an error occurs such that track 701 cannot be read by the system. An entry then is made in the defective tracks/sectors table for that disc pack. The system then attempts rereading the track, and if it can do so successfully, the user's procedure is executed without interruption. If the system cannot read the track, eventually it gives up trying and issues an error message to the user. In either case, when the operator or system supervisor next loads the system, the entry in the defective tracks table is output as an error condition. For example:

```
SUSPECT TRK LDEV#2 CYL 175 HEAD = 1
```

(Assume the above example works out to track 701.) Some reconciliation must be made by the operator or System Supervisor. The Initiator provides four courses of action for the user. They are:

DELETE. This action causes the Initiator to set a permanent "DEFECTIVE" flag within the track entry of the defective tracks/sectors table. The entry is made permanent in the table and cannot be removed. The Initiator then makes a change to the disc free space map such

that all sectors within the specified track cannot be accessed by the system software. If this is not a RELOAD option, the Initiator then causes the file or files (which were stored on the defective track) to be purged.

REASSIGN. This action causes the Initiator to write a special address on the defective track itself. This address is the location of one of the available alternate tracks previously discussed. This action also causes the Initiator to remove the entry from the defective tracks table. If this is not a RELOAD operation, it also causes the file or files which were stored within the defective track to be purged. If this is a RELOAD operation, files can be stored preceding and following the defective track. Figure 6-4 illustrates a reassignment where a new file has been stored in an area with a defective track.

When the file is accessed by a user, the system will read the records from track 700 up to record 191 (see figure 6-4). When records 192 through 383 are requested by the user program, the system will obtain those records from the alternate track. This action will be transparent to the user program. When the user accesses records 384 through the end of the file, the system will obtain them from tracks 702, 703....etc. through the end of the file. This action is not allowed for fixed-head discs.

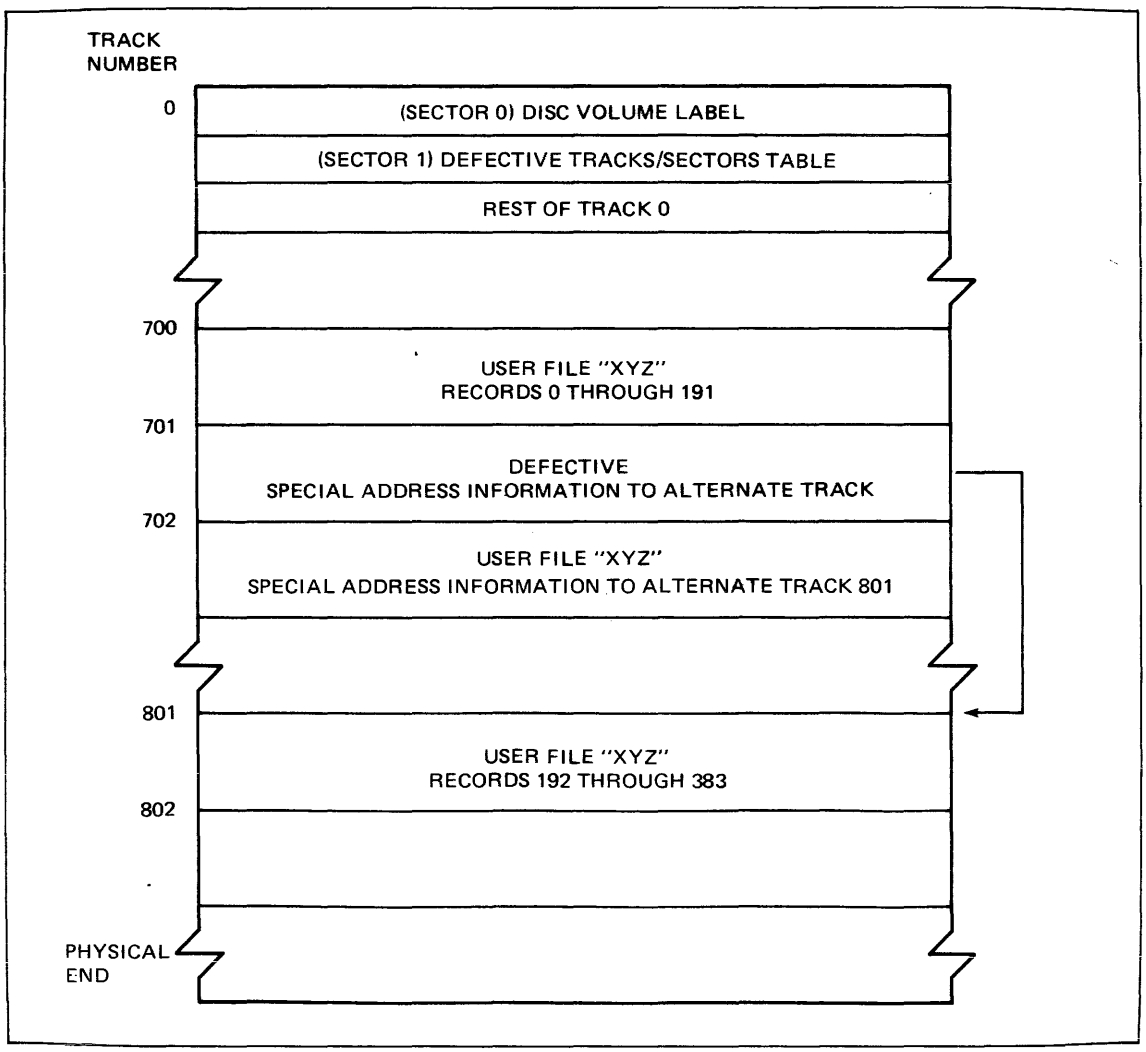


Figure 6-4. Disc With a Track Reassignment

RECOVER. This action causes the initiator program to remove the entry in the defective tracks table only. This means that you are choosing to ignore the suspect-track report. There are several valid reasons for you to do this. For example, a bit of foreign matter may get on the disc surface and cause an error. This option allows you to try to access that track again as if the error never occurred.

CARRIAGE RETURN ONLY. This action on your part leaves the entry in the defective tracks table. When the system is next cold-loaded, the Initiator will again request disposition of the specified track and any new entries which may have been added to the table.

All of the above actions may be done for moving-head discs. For fixed-head discs, the REASSIGN option is prohibited because there are no alternate tracks available. When a moving-head disc contains certain track specific errors, it can be reformatted. Sometimes reformatting will clear the problem on the affected tracks.

Disc errors in certain areas of the physical disc require special considerations:

- Tracks in the directory area of the system disc may not be DELETEd or REASSIGNed except during a RELOAD, in which case the entire directory is moved to a place on the disc large enough for contiguous access to the entire directory. No defective tracks are allowed in the directory area.
- Tracks in the alternate track area of a moving-head disc pack may not be REASSIGNed. The reason for disallowing this is that some hardware controllers can handle only one level of defective tracks for any given access.
- Tracks in the reserved areas of the system disc (or user disc) may not be DELETEd or REASSIGNed. For fast access and chaining of input/output operations, the system software must have contiguous access to certain positions of the disc. For example, the volume label and defective tracks table must always be at sectors 0 and 1 of track 0 on any disc. The bootstrap area of the system disc is another example.
- Tracks in the system area of the disc cannot be DELETEd or REASSIGNed during a COOLSTART. Since this option provides for loading the system from the system disc rather than magnetic tape or serial disc, a track error and subsequent DELETE or REASSIGN would be the equivalent of destroying a portion of the software programs or tables of the system.

In view of the foregoing disc error and recovery function, it can be seen that a regular method of saving files to magnetic tape or serial disc is extremely important.

NOTE

When virtual memory disc space is allocated, the Initiator will try to locate space with no deleted or reassigned tracks. If a large enough contiguous space is not available, it will try again but permitting deleted tracks to be in virtual memory. If the Initiator is still unable to locate space, it will permit re-assigned and deleted tracks in the virtual memory area. If enough space is still not available, then all virtual memory is released from that volume and the message "insufficient disc space" is printed. If the space is found, deleted tracks are removed for the virtual memory available space. If the Initiator detects a suspect track in the virtual area, it will only permit it to be deleted or recovered.

INITIATOR/USER DIALOG

The Initiator/operator or System Supervisor dialog format (Initiator output and user response) is similar to that previously discussed for the :SYSDUMP command.

To begin the dialog, follow the appropriate steps for the Series II/III or Series 30/33/40/44/64:

SERIES II/III SYSTEMS

FOR STARTUP FROM MAGNETIC TAPE

Mount the backup tape on the unit 0 magnetic tape drive and place the device on-line.

On the System Control Panel, set bits 5, 6, 13, and 14 of the SYSTEM SWITCH REGISTER to octal 3006.

While pressing (and holding) the ENABLE switch, press the LOAD switch. This action causes the cold load operation to begin. The Initiator program is read into main memory.

Wait for the RUN light to go out. The SYSTEM HALT light should not be lit.

Press the RUN/HALT switch. The Initiator program is now in execution. If no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32002V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialog by printing the information shown in table 6-3 on the console. (In the dialog shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

If cold loading is done from magnetic tape or serial disc, the Initiator begins its dialog at Step 1.

FOR STARTUP FROM THE HP7976A MAGNETIC TAPE UNIT
VIA THE HPIB INTERFACE MODULE, ON THE SERIES III ONLY

Mount the back-up tape on the unit ∅ magnetic tape drive and place the device on-line.

On the System Control panel, set the lower order bits (Nos. 8-15) on the Switch Register to % 175 (the address of the HPIB Interface). Then set the high order bits (Nos. 0-7) on the Switch Register to the address of the cold load device — the magnetic tape.

While pressing (and holding) the Enable switch, press the LOAD switch. This action causes the cold load operation to begin. The Initiator program is read into memory.

Wait for the RUN light to go out. The System HALT light should not be lit.

Press the RUN/HALT switch. The Initiator program is now in execution. If no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32002V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialog by printing the information shown in Table 6-3 on the console. (In the dialog shown in Table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

Initiator begins its dialog at Step 1.

FOR STARTUP FROM SERIAL DISC OR SYSTEM DISC

Mount the backup serial disc on any drive that does not share a controller with ldn 1 (serial disc may be HP 7902, HP 7905, HP 7906, HP 7920, HP 7925, HP 7933, or HP 9895). If only one controller is available, or if it is desired to place both SDISC and SYSDISC on the same controller, it will be necessary to switch unit numbers during initial, and input the ldn to which the coldload SDISC has been changed. The Initiator will dynamically allocate this ldev to the non-system domain during startup, making it available for private volume use once the system is up.

NOTE

Only the upper (removable) platter of a HP 7905 or 7906 disc drive may be used as a serial disc or private volume. The lower (fixed) platter must be either configured in the system domain or not configured.

On the System Control Panel, set the SYSTEM SWITCH register to the DRT number of the disc drive on which the disc is mounted.

While pressing (and holding) the ENABLE switch, press the LOAD switch. This action causes the cold load operation to begin. The Initiator program is read into main memory.

Press the RUN/HALT switch. The Initiator program is now in execution. If no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32002V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialog by print the the information shown in table 6-3 on the console. (In the dialog shown in the table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

If cold loading is done from serial disc, the Initiator begins its dialogue at step .1. If cold loading is done from the system disc, the dialogue begins with step 0.

SERIES 30 SYSTEMS

FOR STARTUP FROM MAGNETIC TAPE OR A SERIAL DISC

On the System Front Panel, set the LOAD thumbwheel switch to the octal value of the DRT number (channel address and device address) of the device on which the tape or disc is mounted.

On the System Front Panel, press the LOAD; or, the System Console, press the LOAD key while pressing the CNTL key. (Be sure that the console keys are enabled.)

This action causes the cold load operation to begin. The Initiator program is read into main memory.

Press the RUN/HALT switch. The Initiator program is now in execution. After a brief pause, if no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32033V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialogue by printing the information shown in table 6-3 on the console. (In the dialogue shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

The Initiator begins its dialogue at step .1.

FOR STARTUP FROM SYSTEM DISC

On the System Control Panel, set the START thumbwheel switch to the octal valve of the DRT number (device address and channel address) of the system disc. The system disc is always configured as logical device number one (ldn 1).

On the System Front Panel, press the START key; or, on the System Console, press the START key while pressing the CNTL key. (Be sure that the console keys are enabled.)

This action causes the cold load operation to begin. The Initiator program is read into main memory.

The Initiator program is now in execution. After a brief pause, if no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32033V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialogue by printing the information shown in table 6-3 on the console. (In the dialog shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

The Initiator begins its dialog at step 0.

SERIES 33 SYSTEMS

FOR STARTUP FROM MAGNETIC TAPE OR A SERIAL DISC

On the System Front Panel, set the COLD LOAD thumbwheel switch to the octal value of the DRT number (channel address and device address) of the device on which the tape or disc is mounted.

On the System Front Panel, press the COLD LOAD key; or, on the System Console, press the LOAD key while pressing the CNTL key. (Be sure that the console keys are enabled.)

This action causes the cold load operation to begin. The Initiator program is read into main memory.

Press the RUN/HALT switch. The Initiator program is now in execution. After a brief pause, if no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32033V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialog by printing the information shown in table 6-3 on the console. (In the dialog shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

The Initiator begins its dialogue at step .1.

The dialog shown in table 6-3 is similar to that for :SYSDUMP, described previously, and is only explained in detail where the two operations differ.

FOR STARTUP FROM SYSTEM DISC

On the System Control Panel, set the WARMSTART thumbwheel switch to the octal value of the DRT number (device address and channel address) of the system disc. The system disc is always configured as logical device number one (ldn 1).

On the System Front Panel, press the WARMSTART key; or, on the System Console, press the START key while pressing the CNTL key. (Be sure that the console keys are enabled.)

This action causes the cold load operation to begin. The Initiator program is read into main memory.

The Initiator program is now in execution. After a brief pause, if no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32033V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialog by printing the information shown in table 6-3 on the console. (In the dialog shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

The Initiator begins its dialog at step 0.

The dialogue shown in table 6-3 is similar to that for :SYSDUMP, described previously, and is only explained in detail where the two operations differ.

SERIES 40/44 SYSTEMS

FOR STARTUP FROM MAGNETIC TAPE OR A SERIAL DISC

On the System Front Panel, set the LOAD thumbwheel switch to the octal value of the DRT number (channel address and device address) of the device on which the tape or disc is mounted.

On the System Front Panel, press the LOAD key.

This action causes the cold load operation to begin. The Initiator program is read into main memory.

The Initiator program is now in execution. After a brief pause, if no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32033V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialog by printing the information shown in table 6-3 on the console. (In the dialog shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

The Initiator begins its dialog at step .1.

The dialog shown in table 6-3 is similar to that for :SYSDUMP, described previously, and is only explained in detail where the two operations differ.

FOR STARTUP FROM SYSTEM DISC

On the System Control Panel, set the START thumbwheel switch to the octal value of the DRT number (device address and channel address) of the system disc. The system disc is always configured as logical device number one (ldn 1).

On the System Front Panel, press the START key.

This action causes the cold load operation to begin. The Initiator program is read into main memory.

The Initiator program is now in execution. After a brief pause, if no message appears on the system console, press the carriage return on the system console. This is required so that the system hardware will speed sense the console device. The Initiator prints the following message on the console:

HP 32033V.UU.FF

In this message, V is the current MPE version, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialog by printing the information shown in table 6-3 on the console. (In the dialog shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from the user is described in mixed upper and lower-case letters.)

The Initiator begins its dialog at step 0.

The dialog shown in table 6-3 is similar to that for :SYSDUMP, described previously, and is only explained in detail where the two operations differ.

SERIES 64 SYSTEMS

FOR STARTUP FROM MAGNETIC TAPE OR A SERIAL DISC

For magnetic tape, mount the MPE backup tape on the unit 0 magnetic tape drive and place the device on-line. For a serial disc, mount the backup serial disc pack on any drive that does not share the controller on ldev 1. Set the drive to unit 0, being sure no other drives on the same controller are also set to unit 0. (During startup, the Initiator dynamically allocates this drive to the non-system domain, making it available for private volumes use once the startup is complete.)

NOTE

> The instructions given here apply to computers with more than one disc controller. If your system has a serial disc and system disc configured on one controller, read "Standard Instructions," chapter 5 of the Console Operator's Guide, before doing a COLDSTART from serial disc.

If the system is already running, simultaneously press the CNTL key and B (B^C). If the system is not running, press the carriage return. When you see the DCU prompt (C>), enter LOAD on the system console.

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

```
LOAD channel#, device#, imb#
```

Note that once you have overridden the default coldload device in this way, the new values become the default.

When the system quits accessing the coldload medium, the Initiator begins execution and prints:

```
HP 32033V.UU.FF
```

In this message, V is the current version of MPE, UU is the present update-level number, and FF is the fix-level number.

The Initiator next begins its dialog by printing the information shown in table 6-3 on the console. (In the dialog shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from you is described in mixed upper- and lower-case letters.)

The Initiator begins its dialog at step .1.

The dialog shown in table 6-3 is similar to that for :SYSDUMP, described previously, and is only explained in detail where the two operations differ.

FOR STARTUP FROM SYSTEM DISC

If the system is already running, simultaneously press the CNTL key and B (B^C). If the system is not running, press the carriage return. When you see the DCU prompt (C>), enter START on the system console.

When the system quits accessing the coldload medium, the Initiator begins execution and prints the following message:

HP 32003V.UU.FF

In this message, V is the current version of MPE, UU is the present update-level number, and FF is the fix-level number.

At this point there is a one or two-minute delay.

The Initiator next begins its dialog by printing the information shown in table 6-3 on the console. (In the dialog shown in table 6-3, output from the Initiator is shown verbatim in upper-case letters and input from you is described in mixed upper- and lower-case letters.)

The Initiator begins its dialog at step 0.

The dialog shown in table 6-3 is similar to that for :SYSDUMP, described previously, and is only explained in detail where the two operations differ.

Table 6-3. Initiator-User Dialog

STEP NO.	DIALOG
0	WHICH OPTION <WARMSTART/COOLSTART>? Enter the desired option. Note that only the first 3 characters of the option are required. For example WAR or COO. If WARMSTART is selected, the system will recover certain input and all output spoolfiles; unstarted spooled jobs; and partially complete, RESTARTable jobs remaining in the system from the previous execution of MPE. The spoolfiles recovered are available for futher processing. Spooled jobs (i.e., those with spooled job input files) which were executing at the time of interruption will be recovered in the **waiting** state, if you specified RESTART in the :JOB command. All **waiting** spooled jobs are also recovered. **MPE will defer all recovered jobs and output spoolfiles by establishing job and output fences of 14.** (A job or output spoolfile whose priority is at or below the appropriate fence is said to be **deferred**.) If the operator specifies the WARMSTART cold load option, spoolfile recovery is initiated. All output spoolfiles are placed in the READY state, even if they were OPENED or ACTIVE when the interruption occurred. The output fence is set to 14, thereby deferring all output spoolfiles. READY input spoolfiles and spooled, WAITING jobs are also recovered. In order to recover certain EXECUTING jobs, you may specify the RESTART parameter in the :JOB command. A job input spoolfile for a non-RESTARTable job which was executing when the interruption occurred will not be recovered; nor will any :DATA spoolfiles opened by such a job be recovered. All other input spoolfiles open at the time of interruption will be placed in the READY state, completely recovered. Input spoolfiles which were being created (i.e., were ACTIVE) when the interruption occurred can not be recovered and are automatically deleted during a WARMSTART. The job fence is set to 14, thereby deferring all non-HIPRI jobs. These actions are summarized below. **Input Spoolfiles** Interrupted State — WARMSTART Action ACTIVE — Deleted READY — READY (no change) OPENED, not RESTART — Deleted OPENED, RESTART — READY (complete) **Output Spoolfiles** Interrupted State — WARMSTART Action OPENED — READY (incomplete) READY — READY (no change) ACTIVE (single, or last copy) — READY (remaining extents) ACTIVE (m out of n fully copied) — READY (make n-m full copies) Note, however, that recovered **HIPRI** jobs will start immediately; and that **HIPRI** jobs/sessions can be submitted and processed. When the WARMSTART option is selected, the configuration changes portion of the initiator dialog is skipped (i.e., Steps 1 through 68 and steps 70 through 99). If COOLSTART is selected, skip to Step 1. If WARMSTART is selected skip to step 3.1.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
0.1	WHICH OPTION <COLDSTART/RELOAD/UPDATE> ? Enter the option desired. Note that only the first 3 characters of the option are required to be input; for example COL, REL, UPD may be entered for COLDSTART, RELOAD, or UPDATE. All existing spoolfiles are deleted. If RELOAD is chosen and user files exist on the input medium, the Initiator proceeds to Step **0.2** (below). If no user files exist on the input medium, the Initiator skips to step **0.3**. If COLDSTART is selected, the Initiator skips to Step **1**. If UPDATE is selected, the following question is asked. SYSTEM DISC DRT = 4? (Series 33 is 16) If YES press carriage return. If NO, the Initiator skips to Step **1**. NOTE **Files belonging to non-system domain (private) volume sets are not restored during RELOAD.**
0.2	WHICH OPTION <SPREAD/COMPACT/RESTORE/ACCOUNTS/NULL>? Enter the desired RELOAD option, as described below. A carriage return implies SPREAD. The Initiator skips to Step **1**. These options select the algorithm used to determine on which disc a file is placed during RELOAD and whether the account, group, user structure is to be placed into the appropriate directory area, as follows: **SPREAD** An attempt is made to put the file back on a disc in the same device class as it was originally created. If this fails, an attempt is made to replace the file on a disc of the same type and subtype on which it was previously located (e.g., a cartridge disc). If this fails, an attempt is made to put it on a disc of the same type (e.g., a moving-head disc). If this fails, an attempt is made to put the file on any disc in device class DISC. If this fails, a message is printed and the file is not reloaded. In each of these attempts, the files are spread amongst similar discs, if possible. For example, if when the system was dumped there was one cartridge disc which was full, and when it is reloaded there are now two cartridge discs, each will be (approximately) half full. The advantages to SPREAD are disc seeking is reduced since files are spread out and fragmentation is reduced because the disc is repacked. The disadvantage is that if the discs are nearly full it may not be possible to get all files that were previously on the discs back. This situation will be encountered very rarely and when it is, one of the other options may be used. **COMPACT** MPE attempts to place the file back on the same volume from which it came. If this fails, the SPREAD option is used. The major advantage of COMPACT is that if there are no new deleted tracks and the same discs are used, reloading of the system is guaranteed, no matter how full the discs are. In addition, each disc is compacted within the area between deleted tracks (if there are *n* deleted tracks, there will be at most (*n* + 1) fragments). The disadvantage is that the discs may become disparately full.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG	
0.2(Cont'd)	**RESTORE**	MPE attempts to place the files back on the same volume at the same locations from which they came. If this fails, MPE attempts to place the files anywhere on the volume from which they came. If this fails, the SPREAD option is used. The advantages to RESTORE are that it offers the same guarantee made in COMPACT for reloading the system, and that the same files that were previously using alternate tracks are still using them. The disadvantage is that no compacting of the discs is done so that the same fragmentation still exists.
	ACCOUNTS	This option loads the system, creates a directory from the input medium, and loads the system files which reside in the PUB group of the SYS account. No user files are loaded onto the system. The directory which is created will consist of all accounts, groups, and user structures which were current at the time the magnetic tape or serial disc was written by the :SYSDUMP Configurator program. This option is useful because files saved on the input medium by the :SYSDUMP program are compatible with files stored using the :STORE command. In this way you would, for example: 1. Create a new directory structure if the previous directory was destroyed. 2. Conserve vital disc space by selectively loading files into certain accounts with the :RESTORE command after the system is operational.
	NULL	MPE creates a null directory (as described in Step **0.3**) and *no* user files are copied to the disc. NOTE In general, the SPREAD option should be used for reloading the system. If all files cannot be reloaded using this method, COMPACT should be used. RESTORE is useful only when fragmentation is not important but maintaining alternate track assignments is.
0.3	NO USER FILES ON TAPE — DO YOU WANT TO RELOAD? This implies that when the system storage medium was created, no user files were dumped. To proceed with reloading the system, creating a null directory with only the SYS account, PUB group, and MANAGER user, enter YES. To return to Step **0.1** enter NO.	

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
1	ANY CHANGES? To proceed with changes to the current configuration, enter YES. NOTE Any changes you make to the I/O configuration at this step will not be backed up. At your earliest opportunity use SYSDUMP to obtain a back-up for your changes. To change the current configuration, enter YES and proceed to step 2. Otherwise enter NO and skip to step 3.1.
2	LOAD MAP? To request a map on the console showing the correspondences between MPE segments, programs, and code segment table (CST) entries, enter YES. The format of the map is shown in figure 6-5. Note: when a particular running system halts for any reason, it is mandatory to have a LOAD MAP for analysis, provided a system memory dump is taken for the analysis. To suppress this map, enter NO.
3	MEMORY SIZE = <XXX>? The value XXX is the current size of main-memory for the system (in multiples of 1024 words). To change this value, enter one of the following values (specifying memory size in multiples of 1024 words): 64, 96, 128, 160, 192, 224, 256, 384, 512, 768, 1024, 1280, 1536, 1792, and 2048. Series 64 will support up to 4096 x 1024 words.
3.1	At this time, the Initiator confirms that the configured memory size can be supported on the available physical memory. If the configured memory size exceeds the physical memory size, the following message will be output: CONFIGURED MEMORY SIZE EXCEEDS PHYSICAL MEMORY AVAILABLE PHYSICAL MEMORY AVAILABLE IS < MEMSIZE > MEMORY SIZE = <XXX>? If this is a WARMSTART, or if this is a COOLSTART, COLDSTART, UPDATE or RELOAD without changes, skip to step 68.1. Otherwise, proceed to step 4.
4	I/O CONFIGURATION CHANGES? To prepare for addition or deletion of input/output devices, enter YES. To maintain the same input/output device configuration, and proceed to step 68 enter NO.
5	LIST I/O DEVICES? To print a list of input/output devices currently assigned to the system, enter YES. To suppress this listing, enter NO. If an input/output device listing is requested, it is displayed in tabular form. See step 5 of the :SYSDUMP dialog for the format of the listing.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
	NOTE The prompt in step 6, appears only if a communications subsystem device (CS) was previously configured into the system.
6	LIST CS DEVICES? To print a list of the characteristics of all line controllers currently assigned to the system, enter YES. To suppress this listing, enter NO. If a listing is requested, it is displayed in tabular form. See step 6 of the :SYSDUMP dialog for the format of the listing.
7	HIGHEST DRT = <XX>.? XX is a number denoting the present highest DRT entry number that can be assigned to a device. (Note: Do not consider the Multiplexer Channel board DRT when determining this value.) To change XX, enter the new value desired. To maintain the current XX, enter a carriage return.
8	LOGICAL DEVICE #? To specify a device to be added or removed, enter the logical device number of that device. To skip to step 51, enter zero or a carriage return.
9	DRT #? To add a device enter its DRT entry number. If a data communications subsystem was configured into the system, see step 9 of the SYSDUMP dialog. NOTE: For Series 30/33/40/44/64 systems, see notes for step 9, page 6-22. To remove a device and return to step 8, enter zero.
10	UNIT #? Enter the unit number of the device, if the device shares its controller with other devices. NOTE: All ADCC connected terminals must be configured as UNIT 0. If a data communications subsystem was configured into the system, see step 10 of the SYSDUMP dialog. Otherwise, enter zero to continue.
11	SOFTWARE CHANNEL #? If the device is to be on a multicontroller channel, enter channel number, if not, enter zero. Enter zero for all data communications devices and pseudo devices.

Table 6-3. Initiator-User Dialog (Continued

STEP NO.	DIALOG
12	TYPE? Enter the device type (see step 12 of the SYSDUMP dialog).
13	SUB-TYPE? Enter the device sub-type in the range 0 to 15 (see Step 13 of the SYSDUMP dialog). NOTE: Series 40/44 does not support asynchronous half-duplex modems. NOTE If you are configuring a terminal (type 16 or 32, sub-types 14 and 15), the dialog continues to step 14. If you are configuring an HSI (type 19), the dialog skips to step 16. If you are configuring a SSLC (type 18), or an INP (type 17), the dialog skips to step 17. For all other device types, the dialog skips to step 34.
14	TERM TYPE? This question is asked only if type is 16 or 32 (sub-types 14 and 15). To specify default terminal type to be used at log on, enter a value from the list shown to step 14 of the :SYSDUMP dialog. NOTE If term type is not known, enter carriage return.
15	SPEED IN CHARACTERS PER SECOND? This question is asked only if device type is 16 or 32 (sub-types 14 and 15). Specify the terminal speed in characters per second: enter 10, 14, 15, 30, 60, 120, or 240, 360, 480, 960. NOTE The dialog skips to step 34.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
16	PORTMASK? This question is asked only if device type is 19. The values allowable are shown below and must be entered in decimal. This value will form a mask indicating which HSI channel (port) will be part of this device. Only one of the four channels may be designated for each logical device number. Enter 8 HSI cable connector CHAN 0 4 HSI cable connector CHAN 1 2 HSI cable connector CHAN 2 1 HSI cable connector CHAN 3
17	PROTOCOL? Enter 1 for Binary Synchronous Communication.
18	LOCAL MODE? 1 = HSI 1 or 2 = SSLC To define the appropriate mode number for the local station, enter *1* (if local is a primary-contention station) or *2* (if local is a secondary-contention station). NOTE To resolve the problem of contention in point-to-point operation, each station is assigned a priority—primary or secondary. Thus the secondary station can gain control of the line for a transmission only when the line is left free by the primary station. The SSLC is usually a primary station.
19	TRANSMISSION CODE? Enter the appropriate number for the transmission code in use. The code numbers are: 1 = Automatic code sensing of ASCII and EBCDIC if initially receiving; ASCII if initially sending. 2 = ASCII 3 = EBCDIC
	NOTE **For all timeout responses (the next three steps): Entering 0 disables the timeout; maximum timeout is 32000 seconds; the Emulator displays an error when the communications software (CS) disconnects because of a timeout.**
20	RECEIVE TIMEOUT? Enter the positive number of seconds the Communications software will wait to receive text before terminating the read mode. Entering a carriage return provides a 20-second timeout.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
21	LOCAL TIMEOUT? Enter the positive number of seconds a connected local station will wait to transmit or receive before disconnecting. Entering a carriage return provides a 60-second timeout.
22	CONNECT TIMEOUT? Enter the positive number of seconds the local station will wait after one attempt to make a connection to a remote station. Entering a carriage return provides a 900-second timeout.
	NOTE **Step 23 through 25 apply only to CS devices with switched lines connected through a modem. For CS devices with nonswitched lines connected through a modem, the dialog skips to step 26. If the CS is hardwired, the dialog skips to step 28.**
23	DIAL FACILITY? Enter YES when calls can be dialed from the local station. Enter NO when they cannot.
24	ANSWER FACILITY? Enter YES if the local modem can answer calls, either manually or automatically. Enter NO if it cannot. A NO response causes the next step to be skipped.
25	AUTOMATIC ANSWER? Enter YES if the local modem can automatically answer calls. Enter NO if manual answering is required.
26	DUAL SPEED? Enter YES if the local modem is dual speed (European models). Enter NO if it is single speed. A NO response causes the next step to be skipped.
27	HALF SPEED? (This prompt relevant to European models only.) Enter YES if the local modem is to operate at half speed. Enter NO if it is to operate at full speed. The dialog skips to step 29.
28	SPEED CHANGEABLE? Enter YES if the speed of the line is changeable. Enter NO if the line speed is fixed. Note that your response is ignored for modems that provide internal clocking.
29	TRANSMISSION SPEED? For HSI devices, enter 250000 for cable lengths up to 1000'; 125000 for lengths above 1000'. For SSLC's, enter the transmission speed of the line in characters per second (Bit Rate/8).

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
30	TRANSMISSION MODE? Enter the appropriate number for the transmission mode in use. The SSLC may be either half or full duplex, depending upon the type of line and modem. The mode numbers are: 0 = Full duplex (constant carrier) 1 = Half duplex (switched carrier)
31	PREFERRED BUFFER SIZE? Enter the desired buffer size in words, up to a maximum of 4095. Match buffer sizes between sender and receiver if possible.
32	DRIVER CHANGEABLE? Enter NO for HSI or INP NO for SSLC for DSN/DS and 2780/3780 Emulator YES for DSN/MRJE and DSN/MTS
33	DRIVER OPTIONS? Enter 0. The dialog skips to step 43.
34	RECORD WIDTH? Enter the default physical record width (in decimal words) for the device. (See Step 34 of the :SYSDUMP dialog and Appendix A.)
35	OUTPUT DEVICE? If the device can be used as a job or session input device, enter the device class name or logical device number of the device to be used for the corresponding job session listing device. (See step 35 of :SYSDUMP dialog for more details.) If this is not a job session input device, enter zero.
36	ACCEPT JOBS OR SESSIONS? To specify that this device can accept a job or session input stream, enter YES. Otherwise, enter NO or a carriage return.
37	ACCEPT DATA? To specify that this device can accept data directed to a particular job or session input stream, enter YES. Otherwise, enter NO or a carriage return.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
38	INTERACTIVE? To specify that this is an interactive device, enter YES. Otherwise, enter NO or a carriage return.
39	DUPLICATIVE? To specify that this is a duplicative device, enter YES. Otherwise, enter NO or a carriage return.
40	ENABLE SEEKAHEAD? For Series II/III systems with more than one slaved 7906, 7920 or 7925 disc, answer YES to enable seekahead. Otherwise enter NO.
41	INITIALLY SPOOLED? To designate this device as being spooled at cold-load, enter YES. Otherwise, enter NO or a carriage return.
42	INPUT OR OUTPUT? This question is asked only if the device is initially spooled and the device is an input/output device. Enter IN or OUT. Note: Certain input/output devices can be spooled for either input or output. However, no device can be spooled as both input and output at the same time.
43	DRIVER NAME? Enter the name of the program file containing the driver for this device. User program driver names as well as those in Appendices A and B may be used. If driver name is preceded by an asterisk, the driver will be main memory resident at all times.
	NOTE **Steps 44 to 47 apply only to CS devices with switched lines (type 18, subtype 0).** **The dialog for all other devices skips to 48.**
44	PHONELIST? Enter YES to provide a default phone number list. Enter NO if none is to be provided. A NO response causes the next step to be skipped.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
45	PHONE NUMBER? Enter a string of not more than 30 characters.
46	LOCAL ID SEQUENCE? Only the DSN/DS subsystem recognizes local ID sequences. The default local ID sequence can be specified in terms of code or number system. Enter a carriage return for a null local ID sequence. Enter one of the letters below, followed by the ID sequence in quotes, if code, or parentheses, if number system: A = ASCII Example: A "JOE" E = EBCDIC Example: E "STRING" O = Octal Example: O (7, 35, 5) H = Hexadecimal Example: H (A1, 1F, BB) NOTE Do not enter more than 16 characters for the local or remote ID sequence.
47	REMOTE ID SEQUENCE? Only the DSN/DS subsystem recognizes remote ID sequences. Enter the default remote ID sequence in the same format as the local ID sequence (above). This can be repeated until a carriage return is entered.
48	DEVICE CLASSES? Enter a list containing a device class name. Class names are separated from each other by commas. No name need be entered. Only classes DISC and SPOOL (if spooling is desired) are specifically required by the operating system. **If it is not necessary to ask step 49, the dialog returns to step 8.**

Table 6-3. Initiator-User Dialog (Continued

STEP NO.	DIALOG
49	IS classname A SERIAL DISC CLASS? This question is asked (and it is asked for each class being implicitly created by this ldev change) if the device being configured is a supported serial disc device, and classname is not the name of an existing device class. Enter YES or NO. If yes, dialog returns to step 8.
50	IS classname A FOREIGN DISC CLASS? Enter YES or NO. The dialog returns to step 8.
51	MAX # OF OPENED SPOOLFILES = <XXX>.? To change the maximum number of input and output spoolfiles which can be FOPENed at one time, enter the new limit. To retain the current value, enter a carriage return. The highest number you can specify as a maximum is dependent upon the number of logical devices on your system. The algorithm for determining this maximum number of spoolfiles is: $\frac{16244 - (\text{number of logical devices} \times 4)}{30}$ NOTE Each concurrent batch job to be executed requires at least two spoolfiles — one for standard input and one for standard output. Thus, if you specify a maximum of 20 spoolfiles in response to the above prompt, the system is restricted to running ten concurrent jobs.
52	LIST I/O DEVICES? To print a listing of the new input/output device configuration, enter YES. This list appears in the format as described in step 5 of :SYSDUMP dialog. To suppress the list, enter NO or a carriage return.
	NOTE The prompt in the next step appears only if you have configured a CS device into the system.
53	LIST CS DEVICES? Enter YES to list the characteristics of the new CS device configuration. Enter NO to suppress the listing.
54	CLASS CHANGES? To add a class whose preferred order of device allocation can be specified or to add devices to previously defined classes, enter YES. To skip to 66, enter a carriage return.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
55	LIST CLASSES? To list the device classes and the logical devices contained therein, enter YES. See **Step 55 of :SYSDUMP dialog for the format of the list.** To suppress the listing, enter NO or a carriage return.
56	DELETE CLASSES? To delete previously defined classes, enter YES. **Otherwise, enter NO or a carriage return and skip to step 58.**
57	CLASSES? Enter the names, separated by commas, of classes to be deleted.
58	ADD CLASSES? To define new classes or to add devices to previously defined classes, enter **YES. To skip to step 65, enter NO or a carriage return.**
59	CLASS NAME? To define a new class or to add devices to a previously defined class, enter class **name. To skip to step 65, enter a carriage return.**
60	LOGICAL DEVICE #s? Enter logical device numbers, separated by commas, in the preferred order of allocation. **If it is not necessary to ask question 61, then a return is made to step 59.**
61	IN, OUT, OR IN/OUT? Depending upon the types of the devices within the class, this question is asked to determine the desired device class access type. Enter: IN OUT IN/OUT If your response to this question was IN/OUT, then you are asked an additional question:
62	CONCURRENT OR NON-CONCURRENT? That is, are all the devices in the class capable of concurrent IN/OUT? Enter: NC C The dialog returns to step 59.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
63	SERIAL DISC CLASS? This question is asked if all devices in the class are direct-access devices. Enter YES if the classname is to be used exclusively for access of serial devices. Enter NO for all other device classes. NOTE It is possible to configure a disc drive into two classes, one of which is a serial disc class and the other which is not, thus giving the drive the potential for use in either class. **The dialog returns to step 59 if the answer is YES.**
64	**FOREIGN DISC CLASS?** **This question is asked ONLY if the class is NOT a serial disc class.** **Enter YES if the classname is to be used exclusively for access of foreign dics.** **Enter NO if the classname is not foreign disc.**
65	LIST CLASSES? To list the device classes and the logical devices contained therein, enter YES. To suppress the listing, enter NO or a carriage return.
66	LIST I/O DEVICES? To print a listing of the new I/O configuration, enter YES To suppress the listing, enter NO or a carriage return.
	NOTE **The prompt in step 67 appears only if additional drivers are already configured; otherwise, the dialog skips to step 4.**
67	LIST ADDITIONAL DRIVERS? To print a listing showing the presently-configured additional drivers, enter YES. To suppress this listing, enter NO. **The dialog returns to step 4. (Step 4 repeats until it receives a NO response).**

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
68	At this time, one or more of the following **messages** may be output, reflecting errors in the input/output device configuration. If any of these messages appear, the Initiator returns to step 5, unless otherwise noted. DEVICE OF DIFFERENT TYPE RANGES IN CLASS <CLASS> Device types (defined in :SYSDUMP dialog step 4) are divided into ranges from different kinds of devices: **Type Range** — **Class** 0-6 — Direct Access. 7 — Foreign disc. 8-15 — Serial input. 16-23 — Concurrent input/output. 24-30 — Non-concurrent input/output. 31 — Serial disc. 32-39 — Serial output. Within a device ***class***, all defined devices must have types which are compatible; the above message indicates a violation of this rule. See the discussion of device classes (DCATs) in the :SYSDUMP discussion. USER SPECIFIED MAX DRT IS <DRT> FOLLOWING DRTS MUST BE CHANGED LDEV <LDN> DRT <DRT> ... ... At least one device has been defined with a DRT number higher than the specified maximum. Return to step 4. USER SPECIFIED MAX DRT IS <DRT> HIGHEST DRT SUPPORTED BY THIS CPU IS <DRT> The user-specified maximum DRT cannot be supported on this CPU. Return to step 4. USER SPECIFIED MAX DRT IS <DRT> HIGHEST DRT SUPPORTED BY THIS CPU IS <DRT> FOLLOWING DRTS MUST BE CHANGED LDEV <LDN> DRT <DRT> ... ... At least one device has been defined with a DRT number which a) is higher than the specified maximum and/or b) cannot be supported on this CPU. Possibly the user-specified maximum DRT cannot be supported on this CPU. Return to step 4. LOGICAL DEVICE <LDN> DOES NOT EXIST A device (LDN) specified as an output device is not defined in the configuration. LDEV <LDN> AND LDEV <LDN> ON SAME DRT AND UNIT

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
68 (Cont.)	More than one logical device has been defined with the same DRT and unit. NO DEVICE IN CLASS DISC There must always be at least one device defined in the device class DISC. NO OUTPUT DEVICE FOR LOGICAL DEVICE <LDN> Logical device LDN, which has been defined as a device which accepts jobs or sessions, does not have a corresponding job/session list device or device class assigned. OUTPUT CLASS FOR DEVICE <LDN> NO LONGER EXISTS The device class specified as the output (listing device) class for device LDN is not defined in the configuration. SYSTEM TAPE MUST BE DRT 6 UNIT 0 A magnetic tape must be defined for DRT Entry Number 6, Unit 0. If no errors are detected, then the dialog proceeds as follows: if this a RELOAD without changes, skip to step 83. Otherwise, skip to step 69.
68.1	At this time, the Initiator tests to make sure that DRTs have been handled correctly. If any messages appear, the dialog returns to step 4. USER SPECIFIED MAX DRT IS <DRT> FOLLOWING DRTS MUST BE CHANGED LDEV <LDN> DRT <DRT> . . . . . . At lease one device has been defined with a DRT number higher than the specified maximum. USER SPECIFIED MAX DRT IS <DRT> HIGHEST DRT SUPPORTED BY THIS CPU IS <DRT> The user-specified maximum DRT cannot be supported on this CPU. USER SPECIFIED MAX DRT IS <DRT> HIGHEST DRT SUPPORTED BY THIS CPU IS <DRT> FOLLOWING DRTS MUST BE CHANGED LDEV <LDN> DRT <DRT> . . . . . . At least one device has been defined with a DRT number which a) is higher than the specified maximum and/or b) cannot be supported on this CPU. Possible the user-specified maximum DRT cannot be supported on this CPU. If no messages appear, the dialog proceeds as follows: if this is a RELOAD without changes, skip to step 83; otherwise, skip to step 69.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
69	At this time, the Initiator checks to ensure that all volumes defined in the previous cold-load are mounted. If all volumes were not found, the message **FOLLOWING VOLUMES NOT FOUND** is output, followed by a list of volumes that **were defined but not mounted. The Initiator then proceeds to step 70.** NOTE One of the following messages may be printed, indicating that a disc device is not in the ready state: LDEV # <LDN> NOT READY DISC IN DRT # <DRT> UNIT 0 NOT READY You should make the appropriate device ready. It is imperative not to halt the system at this point, (or during any other portion of the Initiator/User Dialogue, unless the RELOAD option was selected), since the Initiator may be in the process of updating the volume labels. If the computer is halted, the message **FOLLOWING VOLUMES NOT FOUND** may be printed during the next cold-load, necessitating a RELOAD operation. If all volumes were found and you had responded YES to the ANY CHANGES? **question in step 1 the Initiator skips to step 72.** If all volumes were found and you had *not* responded YES to the ANY CHANGES? **question in step 1 the Initiator skips to step 83.** **If this is a WARMSTART, control skips to step 100.**
70	LIST VOLUME TABLE? To list the disc volume and their currently-assigned logical device numbers, enter YES; the listing is printed in the following format: **VOLUME #** **NAME** **LOG DEV #** *volnumber* *volname* *ldn* . . . . In this listing, *volname* is a name of up to eight alphanumeric characters, beginning with a letter, identifying the volume; *ldn* is the logical device number assigned to that volume. If *ldn* is 0, this indicates that the volume is not mounted. NOTE **In order to perform a listing of defective tracks (step 79) for any disc, the logical device is required. Listing the volume table will give you the logical devices.** To suppress this listing, enter NO.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
71	MOUNT CORRECT VOLUMES OR RELOAD Following this message, the system halts. The operator should either find the volumes defined on the previous cold load (listed in step 69), mount them, and cold load the system again or cold load using the RELOAD option and redefine the disc volume configuration (as discussed below).
72	DISC VOLUME CHANGES? To prepare for changes to the disc volume configuration or to delete tracks, enter YES. Otherwise, enter NO or a carriage return (control skips to step 87).
73	LIST VOLUME TABLE? To list the disc volume table (as in step 70), enter YES. Otherwise, enter NO. Regardless of whether you requested the listing (by entering YES) or suppressed the listing (by entering NO), at this point in the dialogue, control transfers to step 74 (if this is a RELOAD) or skips to step 76 (if this is a COLDSTART, COOLSTART, or UPDATE).
74	DELETE VOLUME? To prepare to delete a volume, enter YES. To bypass deletion and skip to step 76, enter NO or a carriage return.
75	ENTER VOLUME NAME? To delete a volume, enter the volume name. (When the name is entered, the question is repeated.) Otherwise, enter a carriage return.
76	ADD VOLUME? To prepare to add a volume, enter YES. To bypass addition and skip to step 78, enter NO or a carriage return.
77	ENTER VOLUME NAME To add a volume, enter the volume name. When the name is entered, the question is repeated. Otherwise, enter a carriage return.
78	LIST VOLUME TABLE? To list the disc volumes and their currently-assigned logical device numbers (as in step 70), enter YES. To suppress this listing, enter NO.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
79	LIST DEFECTIVE TRACKS TABLE? or, for 7911, 7912 or 7933 discs, LIST DEFECTIVE TRACK/SECTOR INFORMATION? To prepare for listing the defective tracks table or defective track/sector information which resides on any one of the disc volumes, enter YES. To bypass the listing and skip to step 81, enter NO.
80	LOGICAL DEVICE #? To list the defective tracks table or defective track/sector information for a particular disc, enter the disc's logical device number. (After this listing, the question is repeated). Otherwise, enter a carriage return. If the disc is a type 0 disc, the following information is printed: • Logical size of the device (in cylinders). • Number of alternate tracks available. For each entry in the defective tracks table, the following information is printed: • Cylinder and head number of the defective track. • Absolute sector number of the first sector of the track (in octal). • Absolute sector number of the last sector of the track (in octal). • Track – may be one of the following: (1) SUSPECT – An error has been encountered on this track. (2) SUSPECT ALT – An error has been encountered on the track to which this track was reassigned. (3) UNREADABLE ALT – This track was reassigned to another track but the disc driver was unable to read the alternate track assignment. (4) DELETED – The track is no longer available for use by the system. (5) REASSIGNED – The track has been reassigned to another track. • The cylinder and head number of the alternate track (if the track status is (2) or (5)). If the disc is a type 1 fixed head disc, the following information is provided in the listing for each track in the table: • Track number of the defective track. • Absolute sector number of the first sector of the track (in octal). • Absolute sector number of the last sector of the track (in octal). • Track status – may be (1) or (4) as described above. For type 3 discs, the following information is printed: • Number of spare operations. • Number of spare tracks used. • Number of spare tracks available.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
80 (Cont.)	This is followed by a list of the logically spared tracks on the disc. This includes the logical cylinder and head of the track, and the beginning and ending sector addresses. All tracks will appear as "REASSIGNED." There are no deleted tracks on a type 3 disc. NOTE: The number of spared tracks listed will not necessarily correspond to the "number of spare tracks used." This is because a logical track can be spared more than once, and thereby use up more than one spare track.
81	DELETE TRACK? To prepare for deleting tracks, enter YES. Otherwise, enter NO to skip to step 86. NOTE: If you answer YES and then attempt to delete a track on a type 3 disc, the Initiator will reject the attempt and issue an error message.
82	ENTER LDEV, CYLINDER AND HEAD? NOTE: Listing the defective tracks table of step 79 will provide the input for this dialog. To delete a track on a moving head disc, enter these three parameters, separated by commas: the logical device number of the disc, the cylinder number, and the head number corresponding to the track to be deleted. To delete a track on a fixed head disc, enter the logical device number and track number, separated by a comma. NOTE: The only option available to you when an error occurs on a fixed-head disc is to delete the track, since there are no alternate tracks available. If this is a system disc, see the discussion of disc error recovery for more information. Otherwise, enter a carriage return to proceed to step 84. If input was entered to delete a track, one of the following messages may result: ALTERNATE TRACK – CANNOT DELETE The track is being used as alternate track and cannot be deleted. IN DIRECTORY – CANNOT DELETE The track is in the area used by the directory and cannot be deleted since this is not a RELOAD function. IN DISC FREE SPACE MAP – CANNOT DELETE The track is in the disc free space map and cannot be deleted, since this is not a RELOAD. IN A SYSTEM DISC RESIDENT TABLE – CANNOT DELETE The track is in an area used by a system disc resident table and cannot be deleted, since this is not a RELOAD.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
82 (Cont.)	IN RESERVED AREA – CANNOT DELETE The track is in the area reserved for the disc label or bootstrap program, and cannot be deleted. IN SYSTEM AREA – CANNOT DELETE The track is in the area reserved for the Initiator program and its associated tables, and therefore cannot be deleted since this is a COOLSTART. INVALID CYLINDER NUMBER The cylinder number is not in the correct range for this moving-head disc. INVALID HEAD NUMBER The head number is not in the correct range for this moving-head disc. INVALID TRACK NUMBER The track number is not in the correct range for this fixed-head disc. NOT A DISC The device is not a disc. UNINITIALIZED DISC A volume label has not yet been written on the disc. **WARNING** IN VIRTUAL MEMORY DELETE? The track is in the area reserved for the virtual memory. To delete the track (which will cause the virtual memory to be moved), respond with YES; otherwise respond with NO. Regardless of whether one of these messages printed, control returns to the beginning of this step.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
83	At this point, the Initiator ensures that each disc defined in the configuration has a valid label, that the volume name is defined in the volume table, and that all volumes defined in the volume name is defined in the volume table, and that all volumes defined in the volume table are mounted; the Initiator also lists any suspect tracks, suspect alternate tracks, and unreadable alternate tracks and requests the operator to take action on them. As a result of this verification, one or more of the following messages may appear. (All messages require a response. If no messages appear, the Initiator proceeds directly to step 84.) NOTE: Messages that prompt the user to "DELETE," "REASSIGN," or "RECOVER" tracks will not appear for type 3 discs. (a) INVALID LABEL FOR DEVICE <LDN> ENTER VOLUME NAME? This indicates that LDN does not contain a valid MPE volume label. You must enter a volume name. If the name entered corresponds to that of a volume in the volume table, the logical device number LDN is set for that volume. Otherwise, a new entry is made in the volume table. The disc label is then updated. If device LDN is a type 0 disc, this is followed by the message: LOGICAL PACK SIZE (CYLINDERS) = <SIZE>.? The operator should specify the number of cylinders to be used on this disc; the remainder will be available for alternate track assignments. To retain the default specified by maximum logical size enter a carriage return. Otherwise, enter a value between minimum logical size and maximum logical size as defined by the table below:

Subtype (Appendix A)	Minimum Logical Size	Maximum Logical Size	Maximum Physical Size
0 (Series 33)	65	65	67
2	150	200	203
3	300	400	406
4	300	400	411
5	300	400	411
6	300	400	411
7	90	120	125
8	611	815	823
9	611	815	823
10	300	400	411
11	300	400	411
12	300	400	411

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
83 (Cont.)	(b) DEVICE <LDN> VOLUME <VOLNAME> NOT DEFINED IN TABLE ADD TO SYSTEM VOLUME SET? If you enter YES, the following message is displayed: ENTER VOLUME NAME? This means that the volume identified by VOLNAME, with the logical device number LDN, does not appear in the volume table. To add this volume, identified by this VOLNAME and LDN, enter a carriage return. To add this volume under a different volume name, enter the new name. The new name will be entered in the volume table and the volume will be relabeled with that name. (c) VOLUME NAME <VOLNAME> ON DEVICE <LDN> ALREADY IN USE ENTER VOLUME NAME? This means that two volumes have the same name. To change the name of the volume on device LDN, enter the new name. If the name corresponds to that of a volume in the volume table, the logical device number LDN is set on that volume. Otherwise, a new entry is made in the volume table. The disc label is then updated. (d) ALL VOLUMES MUST BE MOUNTED LIST VOLUME TABLE? This message occurs when an entry appears in the volume table but no corresponding volume is mounted on a disc defined in the configuration. To list the **volume table (as in step 70), response with YES. Otherwise, respond with NO. In either case, control the returns to step 5.** (e) SUSPECT SUSPECT ALT UNREADABLE ALT } TRK LDEV#<LDN> CYL=<CYL> HEAD=<HEAD> This message, applying to a type 0 disc, indicates that an entry in the defective tracks table requires action. (The variable parameters and other related information are described under message (f) below. (f) SUSPECT TRK LDEV#<LDN> TRACK=<TRACK> This message, applying to a fixed-head disc, indicates that an entry in the defective tracks table requires action. The following information applies to both Messages (e) and (f) above. A *suspect track* is the one on which an error has been detected while the system was running. A *suspect alternate track* is one that has been previously reassigned and an error has been detected on the alternate track. An *unreadable alternate track* is one that has been reassigned but the disc driver was unable to read the address of the reassigned track while attempting a transfer. In messages (e) and (f), LDN gives the logical device number of the disc where the error occurred; TRACK gives the track in error for fixed head discs; CYL and HEAD give the cylinder and head number of the bad track for moving head disc.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
83 (Cont.)	Immediately following message (e) or (f), one of the messages (g) through (q) may appear; they require one of the following replies, or YES or NO as indicated: (1) DELETE — To remove the track from the space available for use by the system. (2) REASSIGN — To reassign the space on the bad tracks to one of the available alternate tracks (available only to moving-head discs). (3) RECOVER — To remove the entry from the defective tracks table, ignoring the error. (4) (A *carriage return*) — To ignore the error. The message will be repeated the next time the system is cold-loaded. (g) DELETE OR REASSIGN? The track was flagged as defective previously. Only responses DELETE or REASSIGN are valid. (h) **WARNING** IN ALTERNATE AREA DELETE? The track was flagged as defective previously. It is located in the area reserved for alternate tracks. The only valid response is YES, since tracks in the alternate area cannot be reassigned (reassigned tracks cannot be "chained"). (i) **WARNING** IN ALTERNATE AREA DELETE OR RECOVER? A suspect track is located in the area reserved for alternate track assignment. Valid responses are DELETE, RECOVER or carriage return to ignore the error. (j) FLAGGED TRACK IN RESERVED AREA — MUST REINITIALIZE PACK A track flagged as defective is located in the reserved area of the disc. The system halts at this time. You must either mount a new pack previously formatted or rerun the diagnostic on the present pack to reformat it. Following this, you should restart using the RELOAD option. See Appendix H for formatting procedures. (k) **WARNING** IN RESERVED AREA RECOVER? A suspect track is located in the reserved area of the disc. Respond with YES to remove the entry from the defective tracks table, or NO to leave it in the defective tracks table. Note: RECOVER assumes that the error never occurred. This could be dangerous to system integrity.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
83 (Cont.)	(l) **WARNING** IN SYSTEM AREA RECOVER? A suspect track is located in the area used for the Initiator program and its associated tables. Respond with YES to remove the entry, or NO to leave it in the defective tracks table. (m) **WARNING** IN DIRECTORY RECOVER? A suspect track is located in the area used by the system for the file directory. Respond with YES to remove the entry or NO to leave it in the defective tracks table. (n) **WARNING** IN DISC FREE SPACE MAP REASSIGN OR RECOVER? A suspect track is located in the disc free space map. Enter REASSIGN or RECOVER, or carriage return to ignore the error. (o) **WARNING** IN A SYSTEM DISC RESIDENT TABLE RECOVER? A suspect track is located in a table that is resident on the system disc. Respond with YES to remove the entry or NO to leave it in the defective tracks table. (p) **WARNING** IN VIRTUAL MEMORY DELETE, REASSIGN OR RECOVER? DELETE OR RECOVER? A suspect track is located in the area used for the virtual memory. If this load is not a RELOAD, the second form of the question will be used and responses DELETE or RECOVER are valid. If it is RELOAD using the RESTORE option, the first form of the question will be for moving-head discs and the second form for fixed-head discs. In the case of moving-head discs, responses are DELETE, REASSIGN or RECOVER. For fixed-head discs, only DELETE or RECOVER is allowed. (q) DELETE OR RECOVER? The suspect track is located in no special area of a moving-head disc. All responses DELETE, RECOVER, or carriage return to ignore the problem, are valid. (r) DELETE, REASSIGN OR RECOVER? The suspect track is located in no special area of a moving-head disc. All responses are valid, including a carriage return to ignore the problem. NON-SYSTEM VOLUME ON DEVICE # ADD TO SYSTEM VOLUME SET? Answer YES if the disc volume is to be added to the system domain or NO if it is to be added to the private volumes (non-system) domain.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
84	If any changes have been made to the defective tracks table in step 83, the following question will be printed: LIST DEFECTIVE TRACKS TABLE? or, for 7911, 7912 or 7933 discs, LIST DEFECTIVE TRACK/SECTOR INFORMATION? To prepare for listing the defective tracks table or defective track/sector information of any disc, respond with YES. Otherwise, respond with NO or a carriage return to skip to step 86.
85	LOGICAL DEVICE #? Enter the logical device number of the disc for which the defective tracks table listing is desired (as in step 80). This question is repeated until a carriage return is entered signifying that no further listings are desired.
86	If any changes have been made to the volume table in step 83, the following question is printed: LIST VOLUME TABLE? Respond with YES to obtain a listing of the volume table (as described in step 70); otherwise, enter NO or a carriage return to bypass the listing.
87	If updating from MPE III TO IV, the following message will appear: **WARNING** DEFAULT VIRTUAL MEMORY SIZES BEING USED Therefore, virtual memory is allocated on LDEV1. If virtual memory is desired on other system domain disc, space may be allocated by the following steps.
88	VIRTUAL MEMORY CHANGES? To change the allocation on disc used for virtual memory, enter YES. LDEV1 virtual memory allocation may be changed during RELOAD only; virtual memory allocation on other discs may also be changed during COOLSTART, COLDSTART or UPDATE. To retain the current allocation, enter a carriage return, dialog skips to step 92.
89	LIST VIRTUAL MEMORY ALLOCATION? To list the disc volume, logical device number, and virtual memory allocation, enter YES. To suppress listing, enter a carriage return. VOLUME NAME LDEV # VM ALLOCATION MH 7920U0 1 10

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
90	ENTER VOLUME, SIZE IN KILOSECTORS? (Can enter volume name or LDEV size) LDEV1 can be changed during RELOAD only. You must configure at least 1 kilosector of virtual memory space on LDEV1.
91	VIRTUAL MEMORY CHANGES? Enter a YES, to loop back to step 89. Enter a carriage return to continue dialog.
92	If this system has been configured with the MPE Logging Facility enabled, the following question is printed: DISABLE LOGGING? To turn off all non-mandatory logging, enter YES. Otherwise, respond with NO.
93	MAX # OF SPOOLFILES KILOSECTORS = <XXXXXX>.? To change the maximum number of sectors which can be allocated to spoolfiles (expressed in thousands of sectors), enter the new value. To retain the current limit, enter a carriage return.
94	# OF SECTORS PER SPOOLFILE EXTENT? Enter the size in sectors, for each spoolfile extent. This must be a value between 128 and 32767.
95	If this is not a RELOAD and no tracks have been deleted, the following question is printed: RECOVER LOST DISC SPACE? To recover any disc space that may have been lost because of system failures when temporary files were open, answer YES. Otherwise, respond with NO. NOTE For systems with large numbers of files, this may take between 5 and 10 minutes for every 1000 files.
96	If this load is not a RELOAD and the size of the directory on disc differs from the configured size, the following message will be printed: **WARNING** DIRECTORY SIZE ONLY CHANGED ON RELOAD

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
97	If this load is not a RELOAD and any tracks were deleted or reassigned, the following list may be printed: FOLLOWING FILES PURGED – DISC ERROR *filenname.groupname.accountname* . . . These files reside wholly or partially on tracks that were deleted or reassigned and were therefore purged from the directory. The space used by the files (except for the area on the defective track) was returned to the disc free-space list.
97.1	If any files lost data (but not in the file label) because of tracks or sectors which were spared (reassigned), the following list appears. Otherwise, control skips to step 98. FOLLOWING FILES LOST DATA DURING SPARING . . . The list contains file names and the addresses at which they lost data.
97.2	PURGE ALL FILES WHICH LOST DATA? To purge all files listed in step 97.1 which lost data during sparing, answer YES. This is the recommended response, since the listed files are damaged and probably cannot be relied upon or accessed easily. All of the files are purged and control proceeds to step 98. If a file must be preserved, answer NO and control proceeds to step 97.3.
97.3	SAVE FILE filename.groupname.accountname? This prompt appears for each of the files listed in step 97.1. To save the file, answer YES. To purge the file, answer NO.
98	If this is a RELOAD where user files are to be loaded onto the system, then continue. For all other types of start-up (including RELOAD where no user files are on tape, or the ACCOUNTS option of RELOAD is specified) control proceeds to step 99. The user files read from tape and written to disc can cause one of the following messages to be printed: (a) MOUNT REEL #<REELNUM> The next reel of the set, REELNUM, should be mounted and placed on line. (b) TAPE NOT A MEMBER OF THIS SET The tape mounted in response to message (a) was not a member of the tape set. (c) WRONG REEL Reel REELNUM, specified in message (a), was not mounted. Another reel out of the normal sequence of reels was mounted erroneously.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
98 (Cont.)	(d) NOT A RELOAD TAPE OR IMPROPER TAPE FORMAT The tape mounted in response to message (a) is not a tape generated by :SYSDUMP or :STORE. (e) <XXX> FILES NOT FOUND – ANOTHER TAPE SET AVAILABLE? Not all of the files in the directory when the system was dumped have been found on the tape. If another set of tapes is available, respond with YES, mount the first reel of the set, and place it on line. Otherwise, respond with NO. Note: When the logging facility is enabled, the most recent in-use log file is not dumped to magnetic tape, because it is busy. However, the directory entry for the file is dumped to the tape. On RELOAD, the above message may appear even if the entire user file set is loaded onto the system. The directory entry for the log file plus the actual absence of the file on tape will provoke this message. Of course, there may be one or more other files which are missing from the tape. If you answer NO, it is advisable to list the files not found when the question is asked in the dialog below. (f) WRONG TAPE SET – MUST HAVE EARLIER DATE The tape mounted in response to message (e) has a date later than the previous tape set processed. The operator should find the first reel of the correct set, mount it, and place it on line. (g) NOT ALL FILES FOUND – LIST? To list files not found on reload tape, enter YES. To suppress the listing and skip to step 11, enter NO or carriage return. Note: Listing a large number of "files not found" on the system console device can be very time consuming. (h) FOLLOWING FILES PURGED BECAUSE OF INSUFFICIENT DISC SPACE (1) TAPE PARITY ERROR (2) LIST? To list files which were found on reload tape but for which disc space was unavailable, enter YES. To suppress the listing, enter NO or carriage return. See note in step (g) above. (Note: the message, NAME.IS.UNKNOWN, received during the listing of files, means the parity error occurred while reading the file label.) (i) DISC SPACE ERROR The Initiator erred while attempting to return allocated disc space after detection of a parity error on tape. This halts the configuration. Dialog changes to Initiator-User for recovery from parity errors during RELOAD of user files. (j) **WARNING** DOUBLE EOF MARK CONTINUE FROM NEXT REEL? Two consecutive EOF marks were read. Either the tape is of the wrong format or a parity error was detected during reading of the tape trailer label. (It should be possible to determine if the tape is located at the trailer label by the position of the tape reel. If this is the case enter YES, otherwise, enter NO. The Initiator cannot recover from improper tape format. A parity error in the trailer label will cause the first file of the next tape to be skipped, (probably a continuation of the last file of the previous tape) and continue RELOAD with the next file.

Table 6-3. Initiator-User Dialog (Continued)

STEP NO.	DIALOG
98 (Cont.)	(k) IRRECOVERABLE PARITY ERRORS Initiator encountered parity errors in both the header and trailer levels of the tape and was not able to continue with the RELOAD operation. The Initiator aborts as a result.
99	One of the following messages may be printed due to disc errors: (a) **WARNING** OLD WELCOME MESSAGE ON DEFECTIVE TRACKS If WARMSTART, COOLSTART, or UPDATE, an attempt is made to recover welcome message but if it or DST Descriptor table is located on defective tracks, welcome message is set to null. Operator should enter new welcome message after load is complete. (b) CANNOT RECOVER DISC SPACE OF SPOOLFILE #XXXX A disc error occurred during the attempt to recover lost disc space of the input/output spoolfile.
99.1	BANK 0 DEPENDENT MEMORY USED – XXXXXX (up to 64K) (amount of bank 0 memory used during MPE system initialization)
100	DATE (M/D/Y)? Enter a carriage return or the current date in the following format: *mm/dm/yr* where *mm* = One or two digits representing the month. *dm* = One or two digits representing the day of the month. *yr* = The last two digits of the year. If a carriage return is entered, the next question is skipped and MPE prints: WED, NOV 1, 1972, 12:00 AM
101	TIME (H:M)? Enter the current time of day in the following format: *hh:mm* where *hh* = One or two digits indicating the hour (on a 24-hour basis). *mm* = One or two digits indicating the minute.
102	MPE Prints the date and time and the message? (Y/N) If the date and time were entered in steps 100 and 101, MPE echoes them for verification. If you respond with NO (or N), the Initiator returns to step 100. If the date and time are correct, enter YES or carriage return to continue.
The Initiator program now terminates, transferring control to MPE. When the system prints *WELCOME*, it is ready for use.	

LOAD MAP

A typical map requested in Step 2 of the Initiator/user Dialog appears as shown in figure 6-5. This map shows the correspondence between MPE code segments and programs, and code segment table (CST) entries. First, the CST number (in octal) is shown. Next, the System Segmented Library (SL) segment name or program name is presented. SL segment names are followed by a parenthesized value, program file names are not. The parenthesized number indicates the logical segment number of the segment within the system library, identified as SL.PUB.SYS.

```
01 ININ                    31 ALLOCATE (27)
02 FILESYS1 (0)            32 ALLOCUTIL (30)
03 FILESYS4 (1)            33 HARDRES (31)
04 FILESYS5 (2)            34 ABORTDUMP (32)
05 FILESYS6 (3)            35 MESSAGE (33)
06 FILESYS6A (4)           36 PROCSEG (34)
07 FILESYS7 (5)            37 NRIO (35)
10 CIALTORG (6)            40 PCREATE (36)
11 CICOMSYS (7)            41 MORGUE (37)
12 CIERR (10)              42 BIPC (40)
13 CIFILEB (11)            43 IPC (41)
14 CIFILEM (12)            44 CHECKER (42)
15 CIINIT (13)             45 UTILITY1 (43)
16 CILISTF (14)            46 UTILITY2 (44)
17 CIMISC (15)             47 LOADER1 (45)
20 CIORGMAN (16)           50 RINS (46)
21 CIPREPRUN (17)          51 JOBTABLE (47)
22 CISUBS (20)             52 DEBUG (50)
23 CISYSMGR (21)           53 NURSERY (51)
24 CIUSERUTIL (22)         54 SPOOLING (54)
25 CXSTOREST (23)          55 SPOOLCOMS1 (55)
26 RESTORE (24)            56 SPOOLCOMS2 (56)
27 STORE (25)              57 PVCOMSEG (57)
30 DIRC (26)               60 PVSYSD (60)
```

Figure 6-5. Load Map

```
 61 PVSYSM (61)
 62 UDC (62)
 63 USER (63)
 64 HELPUSER (64)
 65 OPLOW (65)
 66 OPMED (66)
 67 OPHI (67)
 70 LABSEG (70)
 71 SDISC (71)
 72 LOGSEG0 (73)
 73 LOGSEG1 (74)
 74 KERNELC (75)
 75 KERNELD (76)
 76 MISCSEGC (77)
 77 FILESYS1A (101)
100 FILESYS2 (102)
101 FILESYS3 (103)
102 DEGUGUTIL (104)
103 SEGUTIL (105)
104 KSAM01 (106)
105 KSAM02 (107)
106 KSAM03 (110)
107 KSAM04 (111)
110 KSAM05 (112)
111 FIRMWARESIM1 (52)
112 FIRMWARESIM2 (53)
113 KSAM06 (113)
114 KSAM07 (114)
115 COMSYS1 (116)
116 COMSYS3 (120)
117 COMSYS4 (121)
120 COMSYS5 (122)
121 CSUTILTY (123)
122 COMSYS2 (117)
123 BSCLCM (124)
124 BSCSLCP0 (125)
125 DVRSSLC (126)
126 DVRHSI (127)
127 DSSEG1 (151)
130 DSSEG2 (152)
131 DSSEG4 (154)
132 DSMISC (156)
133 DSIOM (157)
134 DSSEG3 (153)
135 DSSEG5 (155)
136 CLIB'01 (204)
137 CLIB'03 (206)
140 CLIB'04 (207)
141 CLIB'05 (210)
142 DSRTECALLS (160)
143 MRJEMISC1 (161)
144 MRJEMISC2 (162)
145 MRJESLCP (163)
146 BSCSLCP1 (164)
147 MPMONCMD (165)
150 IMAGE01 (214)
151 IMAGE02 (215)
152 IOMONITOR3270 (231)
153 TRACE0' (232)
154 TRACE1' (233)
155 IOMDISC1
156 IOTEPE0
157 IOTAPE0
160 IOTERM0
161 IOLPRT0
```

Figure 6-5. Load Map (Continued)

COLD LOAD ERROR MESSAGES

One of the error messages shown in table 6-4 may appear while cold loading the system. Following the printing of the message, the machine halts.

Table 6-4. Cold Load Error Messages

MSG #	MESSAGE	COMMENT
1	NON-RESPONDING DEVICE CONTROLLER DRT <DRT>	An I/O instruction returned condition code of CCL meaning the device controller did not respond. This could be due to a configuration problem. If the I/O configuration is correct, contact the HP Customer Engineer.
2	CHANNEL PROGRAM FAILURE – DRT <DRT>	An I/O instruction returned a condition code of CCL or CCG. This could be due to a configuration problem. If the I/O configuration is correct, contact the HP Customer Engineer.
3	CHANNEL PROGRAM ABORTED – CPVA %<STATUS>	%140000 – DMA abort %164000 – Channel program aborted due to an address rollover. %162000 – Channel program aborted due to an HP-IB parity error. %161000 – Channel program aborted due to a status change in the HP-IB interface. %160400 – Channel program aborted due to a device clear during channel program. %160200 – Channel program aborted due to a FIFO handshake abort. %160100 – Channel program aborted due to a serial poll error. %160040 – Channel program aborted due to an illegal CSRQ. %160020 – Channel program aborted due to a memory parity error. %160010 – Channel program aborted due to a non-responding IMB module. %160004 – Channel program aborted due to a channel hardware timeout. %160002 – Channel program aborted due to a data chain error. %160001 – Channel program aborted due to an invalid instruction.
4	IOP ERROR	An impossible status indication was returned by the tape controller, indicating a failure in the controller or the input/output processor. At this time it is advisable to request assistance from your HP Customer Engineer. It is possible to try the requested cold-load operation again. The system may commence normal operations, but could be extremely unreliable. File integrity could be lost. It is better to have the Customer Engineer check out the system before you try again.

Table 6-4. Cold Load Messages (Continued)

MSG #	MESSAGE	COMMENT
5	TAPE I/O CMD REJECTED	This error, reported by the magnetic tape controller, implies that there is a bad tape controller. If you have more than one drive on your system, check to ensure that only one drive on-line and the proper unit has been selected (unit 0). Retry the operation; if it fails with the same error message, then contact your HP Customer Engineer.
6	TAPE TRANSFER ERROR	This error, reported by the magnetic tape controller, implies either a defective tape or a defective tape controller. Try a back-up tape after cleaning the tape heads.
7	TAPE PARITY ERROR	A parity error was detected while reading the magnetic tape. Try a back-up tape after cleaning the tape heads.
8	TAPE TIMING ERROR	This error, reported by the magnetic tape controller, implies a defective tape controller. Contact your Hewlett-Packard Customer Engineer.
9	READING BLANK TAPE	This error, reported by the magnetic tape controller, implies either a defective tape or a defective tape controller. Try a different magnetic tape back-up of the system after cleaning the tape heads. If that fails and the same error is printed, contact your HP Customer Engineer.
10	COLD LOAD TAPE READ ERROR	A tape-read error was detected during the cold load operation. This may be a magnetic tape hardware problem. Clean tape heads and try again. If error persists, temporarily use an *obsolete* system back-up tape and try to load. If this fails, contact the HP Customer Engineer immediately.
11	IRRECOVERABLE PARITY ERRORS	This error, reported by the magnetic tape controller, implies either a defective tape or a defective tape controller. Try a different magnetic tape back-up of the system after cleaning the tape heads. If that fails and the same error is printed, contact your HP Customer Engineer.
12	ILLEGAL CMD	This is an error returned by the disc controller.
13	CYLINDER COMPARE ERROR	This is an error returned by the disc controller.
14	UNCORRECTABLE ERROR	This is an error returned by the disc controller.

Table 6-4. Cold Load Error Messages (Continued)

MSG #	MESSAGE	COMMENT
15	HEAD SECTOR COMPARE ERROR	This is an error returned by the disc controller.
16	SIO PROGRAM ERROR	This is an error returned by the disc controller.
17	END OF CYLINDER	This is an error returned by the disc controller.
18	OVERRUN	This is an error returned by the disc controller.
19	POSSIBLE CORRECTABLE ERROR	This is an error returned by the disc controller.
20	ILLEGAL ACCESS TO SPACE TRACK	This is an error returned by the disc controller.
21	DEFECTIVE TRACK	This is an error returned by the disc controller.
22	HEAD MOVING DURING OPERATION	This is an error returned by the disc controller.
23	DISC DRIVE ERROR	This is an error returned by the disc controller.
24	ATTEMPT TO PROTECT DEFECTIVE TRACK	This is an error returned by the disc controller.
25	DRIVE UNAVAILABLE	This is an error returned by the disc controller.
26	DRIVE ATTENTION	This is an error returned by the disc controller.
27	BAD DISC ADDRESS	An address greater than the available number of sectors on the disc was passed to the disc driver. This usually indicates an error internal to the system.
28	BAD FILE ADDRESS	An attempt was made to write outside the range of one of the system files. A RELOAD should be attempted. If the system disc is insufficient in size to hold all system files, this error will occur again. If another, larger capacity disc is available, use it as the system disc.
100	PREVIOUS RELOAD ABORTED – MUST RELOAD	The last coldload was RELOAD that was aborted. Therefore, coldload must be a RELOAD.

Table 6-4. Cold Load Error Messages (Continued)

MSG #	MESSAGE	COMMENT
101	PREVIOUS TAPE COLD LOAD ABORTED – MUST COLD LOAD FROM TAPE	The last coldload was a COLDSTART or UPDATE was aborted. Therefore, this coldload must be a COLDSTART, UPDATE or RELOAD.
102	USER SPECIFIED MAX DRT IS <DRT>	At least one device has been defined with a DRT number higher than the specified maximum. Increase the value for "MAXDRT".
103	SYSTEM DISC MUST BE UNIT 0	
104	SYSTEM DISC MAY ONLY BE RECONFIGURED ON RELOAD	The type and subtype on the disc label do not match the type and subtype of the system disc in the I/O configuration.
105	LDEV 1 MUST BE SYSTEM DISC	
106	WRONG DRT FOR SYSTEM DISC	This error happens only during an UPDATE. This disc label is incorrect for a system disc, the disc is not on-line, or the device is not a disc.
107	NO DEVICE IN CLASS DISC	At least one system disc must be in class "DISC".
108	COLD LOAD DEVICE MUST BE CONFIGURED	
109	SYSTEM CONSOLE MUST BE IN DRT 7 UNIT 0	The system console must be DRT 7 unit 0 for Series II/III.
110	SYSTEM CONSOLE MUST BE IN DRT 8 UNIT 0	The system console must be DRT 8 unit 0 for Series 30/33/40/44/64.
111	IMPROPER SPEED FOR SYSTEM CONSOLE	
112	LDEV <LDEV> TERMINAL UNIT MUST BE 0	The Series 30/33/40/44/64 terminal controller does not support multiple units.
113	LDEV <LDEV> TERMINAL DRT MUST BE UNIQUE	The Series 30/33/40/44/64 terminal controller does not support multiple units.
114	ILLEGAL MASTER DEVICE	Specified value is not the logical device number of a configured communications interface (DSN/INP, SSLC or HSI port). When configuring a communications driver or virtual terminal you must assign the logical device number (preceded by a pound sign) of the associated communications interface.
115	UNDEFINED CLASS <CLASS> USED AS OUTPUT DEVICE BY FOLLOWING DEVICES	
116	NO OUTPUT DEVICE FOR LOGICAL DEVICE <LDEV>	See Initiator-User Dialog, step 68.
117	OUTPUT CLASS FOR DEVICE <LDEV> NO LONGER	See Initiator-User Dialog, step 68.

Table 6-4. Cold Load Error Messages (Continued)

MSG #	MESSAGE	COMMENT
118	DEVICE CLASS <CLASS> CANNOT BE OUTPUT DEVICE	
119	LOGICAL DEVICE <LDEV> CANNOT BE OUTPUT DEVICE	
120	LOGICAL DEVICE <LDEV> DOES NOT EXIST	See Initiator-User Dialog, step 68.
121	LDEV <LDEV> AND LDEV <LDEV> ON SAME DRT AND UNIT	See Initiator-User Dialog, step 68.
122	DEVICES OF DIFFERENT TYPE RANGES IN CLASS <CLASS>	See Initiator-User Dialog, step 68.
123	ILLEGAL TYPE COMBINATIONS IN CLASS <CLASS>	See Initiator-User Dialog, step 68.
124	CONFIGURED MEMORY SIZE EXCEEDS PHYSICAL MEMORY AVAILABLE	
125	CANNOT RECOVER DISC SPACE OF SPOOLFILE <NUM>	
126	DISC DRIVER DOES NOT EXIST	A transfer has been attempted to a disc with a type or subtype not known to the system. This usually indicates an error internal to the system. If you have inserted user written drivers into the system, try deleting them on a RELOAD. If this fails, try a temporary RELOAD from an obsolete system back-up tape. If this fails, contact your HP Customer Engineer.
127	MORE THAN ONE DEVICE FOR DRT <DRT>	
200	VOLUME TABLE FULL	You have tried to configure more than 64 volumes.
201	VOLUME TABLE DESTROYED – MUST RELOAD	The volume table maintained on the system disc has been overwritten; the system must be reloaded.
202	MOUNT CORRECT VOLUMES OR RELOAD	The volumes specified in the volume table do not agree with the mounted volumes.
203	ALL VOLUMES MUST BE MOUNTED	The volumes specified in the volume table cannot be found among the mounted discs.
204	DEVICE <LDEV> VOLUME <VOLUME> NOT DEFINED IN TABLE	A system volume is mounted on this ldev that is not declared in the volume table.
205	VOLUME NAME <VOLUME> ON DEVICE <LDEV> ALREADY IN USE	Two volumes are mounted with the same name. You will be requested to assign the volume a new name.
225	DEFECTIVE TRACKS TABLE FULL	The table can hold a maximum of 120 deleted or resigned tracks.

Table 6-4. Cold Load Error Messages (Continued)

MSG #	MESSAGE	COMMENT
226	NO ALTERNATES AVAILABLE	There are no alternate tracks available to reassign a track.
227	IN RESERVED AREA – CANNOT DELETE	This track is in either the first 400 sectors of the disc or the disc free space table.
228	IN DIRECTORY – CANNOT DELETE	
229	ALTERNATE TRACK – CANNOT DELETE	This alternate track is the target of a re-assigned track and therefore cannot be deleted.
230	IN SYSTEM AREA – CANNOT DELETE	
231	SYSTEM TABLE ON DEFECTIVE TRACKS – CANNOT WARMSTART	At least one of the tables – DST Descriptor table, JMAT, IDD or ODD – which must be recovered for a WARMSTART is located on defective tracks. Either COOLSTART, COLDSTART, or RELOAD may be attempted. The Initiator will then lead you to the defective track as outlined in the dialogue. You will be requested to take some action on the defective track.
232	FLAGGED TRACK IN RESERVED AREA – MUST REINITIALIZE PACK	A defective track is illegal in the first 400 sectors of the disc or in the disc free space table.
250	TERMINAL INITIALIZATION PROGRAM TOO LARGE	This usually indicates an error internal to the system. Contact your HP Customer Engineer.
251	TERMINAL CHANNEL PROGRAM TOO LARGE	This usually indicates an error internal to the system. Contact your HP Customer Engineer.
252	DIFFERENT PROCESSES SPECIFIED FOR DRT <DRT>	This usually indicates an error internal to the system. Contact your HP Customer Engineer.
253	INITIATOR, COMPLETOR FOR LDEV <LDEV> SUPPOSED TO BE CORE RESIDENT.	This usually indicates an error internal to the system. Contact your HP Customer Engineer.
254	MORE THAN ONE PRIMARY INTERRUPT HANDLER FOR DRT <DRT>	This usually indicates an error internal to the system. Contact your HP Customer Engineer.
275	INVALID DIRECTORY ENTRY – ENTRY LEVEL WRONG OR MISSING AT ACCOUNT, GROUP OR FILE LEVEL	Bad directory – reload.
276	INVALID DIRECTORY ENTRY – ENTRY LEVEL WRONG OR MISSING AT ACCOUNT OR USER LEVEL	Bad directory – reload

Tabe 6-4. Cold Load Error Messages (Continued)

MSG #	MESSAGE	COMMENT
277	DIRECTORY ERROR A= <A>, B= <B>	An error occurred while accessing the directory. The directory software, which is a part of the operating system, is also included in the Initiator in slightly modified form. The errors which occur for the Initiator are the same type of errors that can occur to a user when the system is operational. The initiator cannot, however, recover these errors because it does not have the same sophistication as the operating system has. All of the possible errors which the directory software can report are listed below. Some of the errors are not possible in the Initiator because they are illogical at this point. A = 1 Duplicate Filename Detected — B = Non Relevant A = 2 Non-Existent Name at Some Point in the Search — B = The Non-Existent Node in the Form of a Number as Follows: 0 = FILE, 1 = GROUP, 2 = ACCOUNT, 3 = USER *A = 3 User Does Not Have "Save" Access — B = The Access Level to Which User Does Not Have Access in the Form of a Number as Follows: 1 = GROUP, 2 = ACCOUNT A = 4 No Room. Cannot Accommodate Date any More Entry Blocks in Directory — B = Percentage of Total Entry Space Actually in Use. *A = 5 No Room. More Than 65K Entries in Directory — B = Not Relevant A = 6 No Room. Cannot Accommodate More Contiguous Blocks in Directory — B = Number of Contiguous Blocks Being Requested *A = 7 Entry Cannot be Purged Because it is Busy — B = Not Relevant *A = 8 File Space Limit Exceeded — B = The Permanent File Space Requested has Been Exceeded for the Node in the Form of a Number as Follows: 1 = GROUP, 2 = ACCOUNT * *Invalid error conditions for the Initiator only.*

Table 6-4. Cold Load Error Messages (Continued)

MSG #	MESSAGE	COMMENT
300	OUT OF CST ENTRIES	The Initiator has run out of code segment table entries for the system. Re-build the system tape (through the :SYSDUMP command.), increasing the number of entries in this table (step 4.1 of the Configurator/ User Dialog).
301	OUT OF DST ENTRIES	The Initiator has run out of data segment table entries for the system. Re-build the system tape (through the :SYSDUMP command), in-creasing the number of entries in this table (step 4.3 of the Configurator/ User Dialog).
302	OUT OF PCB ENTRIES	The Initiator has run out of process control block table entries for the system. Re-build the system tape (through the :SYSDUMP command), increasing the number of entries in this table (step 4.4 of the Configurator/ User Dialog).
303	OUT OF WSTAB ENTRIES	The Initiator has run out of working set table entries for the system. Re-build the system tape (through the :SYSDUMP command), increasing the maximum number of concurrently running programs allowed on the system (step 9.1 of the Configurator/User Dialog).
304	OUT OF CSTBLK ENTRIES	The Initiator has run out of extended cst table entries for the system. Re-build the system tape (through the :SYSDUMP command), increasing the number of entries in this table (step 4.2 of the Configurator/ User Dialog).
305	INVALID JMAT DESTROYED BEFORE WARMSTART DESTROYED DURING WARMSTART	The JOB MASTER TABLE has been destroyed. Coolstart from disc.
325	DISC SPACE ERROR	A conflict exists between the disc free-space map and the space used as defined in the directory. A RELOAD should be attempted. If RELOAD fails, attempt to format all packs and then attempt RELOAD. If that fails, contact your HP Customer Engineer.

Table 6-4. Cold Load Error Messages (Continued)

MSG #	MESSAGE	COMMENT
326	OUT OF DISC SPACE ON LDEV <LDEV>	See comment for # 325
327	OUT OF BOOTSTRAP DISC SPACE	The 400 sectors of the system disc allocated to the bootstrap program and certain tables has been exceeded; this is an error internal to the system. A RELOAD should be attempted. If that fails, contact your HP Customer Engineer for assistance.
328	DISC SPACE ERROR – WHILE RESTORING SPACE	See comment for # 325
329	DISC SPACE ERROR – WHILE GETTING SPACE	See comment for # 325
330	OUT OF VIRTUAL MEMORY ON LDEV <LDEV>	
350	OUT OF MEMORY	The total amount of main-memory needed to build the resident portion of the system has been exceeded. You can decrease the size of core resident tables through SYSDUMP (See Appendix C of System Manager/System Supervisor). This error can also be caused by too many core resident code segments, although unlikely.
351	OUT OF BANK Ø	Core resident tables that need to be resident in the first 64K words (128K bytes) of memory, exceed bank Ø. You can delete some I/O devices or decrease the size of core resident tables through SYSDUMP. See Appendix C of System Manager/System Supervisor.
352	BANK WRAP-AROUND	You configured too many devices during the I/O configuration. You should delete unneeded devices during the I/O configuration.
374	FATAL ERROR <NUM>	This usually indicates an error internal to the system. Contact your HP Customer Engineer.
375	IMPROPER TAPE FORMAT	The information on a tape used for RELOAD does not agree with the format of tapes produced by the :STORE and :SYSDUMP COMMANDS. Double check all reels of tape to ensure they are properly marked as :STORE or :SYSDUMP tapes.
376	NOT A RELOAD TAPE	The tape label is either not 40 words long or the label identification is incorrect. Mount the correct tape and proceed.

Table 6-4. Cold Load Error Messages (Continued)

MSG #	MESSAGE	COMMENT
377	TAPE NOT A MEMBER OF THIS SET	The creation date in the tape label is incorrect for this tape to be a member of this reload set. Mount correct tape and proceed.
378	WRONG TAPE SET – MUST HAVE EARLIER DATE	The creation date in the tape label exceeds the date of the reload. Mount an earlier dated tape and proceed.
379	WRONG REEL	The reel identification in the tape level does not match the expected reel to be mounted. Mount the correct reel and proceed.
400	NON-RESPONDING MODULE INTERRUPT	This error can occur from referencing a non-existent I/O channel of memory controller. Check your configuration of I/O devices for invalid DRTs and your configured memory size. If configuration is correct, contact the HP Customer Engineer.
450	FILE LABEL CHECKSUM ERROR	This error message is generated when a system file has a file label checksum error.
451	TAPE FILE LABEL CHECKSUM ERROR	This error message is generated when the new system file from tape/serial disc has a file label checksum error.
452	DEFECTIVE FILE LABEL	This error message is generated whenever there is a volume number in the file label that does not exist on the system.

COLD LOAD ERROR MESSAGES WITHOUT NUMBERS

DISC READ / WRITE / SEEK ERROR ON LDEV# <LDEV> STATUS= %<STATUS> ADDR=%<ADDR> WORDS=<WORDS>	A disc error has occurred on the specified logical device. The user should cold-load again, as he will be prompted to take action on the bad track.
EOF	An attempt has been made to read past the end of one of the system files. Try an UPDATE or COLD LOAD from tape. If this fails a RELOAD should be attempted. If this fails, temporarily use an obsolete system back-up tape in a RELOAD. If that fails, contact your HP Customer Engineer.
FILE<NAME>.PUB.SYS NOT ON DISC	The specified file was needed but was not found in the directory. Obtain another tape containing the system and UPDATE. If no current system tape is available, attempt UPDATE to an obsolete version of the system temporarily, then contact your HP Customer Engineer for a current copy of the system.
TAPE UNIT WENT NOT READY	This error, reported by the magnetic tape controller, implies either that the tape controller is defective or you switched the tape unit off-line during an operation.

BACKING UP DISC FILES ON A SERIAL STORAGE MEDIUM

MPE provides two methods of backing-up files. These are:

1. The :SYSDUMP command, used to dump all files on the system or only those files most recently changed, and the current directory and accounting information.

2. The :STORE command, defined in Section II. This command is available to standard users in order to back up those files to which the user has read access. System Manager and System Supervisor users, however, have the ability to dump any or all files in the system by using the :STORE command.

Files produced by :SYSDUMP and :STORE are compatible. Those written by :SYSDUMP are used by the MPE Initiator when reloading the system (RELOAD option). Files produced by either method are suitable as input to the :RESTORE command (described in Section II) which allows a System Manager or System Supervisor to restore to disc from a back-up medium any file in the system.

In general, :SYSDUMP should be used for daily back-up of the system (with the options previously discussed), since it provides a record of the latest accounting information. :STORE/ :RESTORE is used when it is desirable to back-up only those files which belong to a particular set of groups or accounts. However, :STORE/:RESTORE can be used by the System Manager or System Supervisor to save or load any or all files on the system provided the appropriate accounts, group, and user structures already exist.

MAGNETIC TAPE FORMAT

Tapes produced by the :STORE command may be labeled or unlabeled. Unlabeled :STORE tapes are compatible with those produced by the :SYSDUMP command. Tapes produced by either :STORE or :SYSDUMP are suitable as input to the :RESTORE command.

The general formats of labeled and unlabeled magnetic tapes created by the :STORE command are presented in figure 6-6. These formats are defined in greater detail in Appendix F. Both :STORE and :RESTORE support multifile and multi-reel files.

The *tape directory* records are 12 word records with a default *blocksize* of 4096 words. There is one entry for each file on the tape. The entries are ordered the same as the files on the tape. See Appendix F for further information.

The *recsize* parameter of the :FILE command can be used to change *record size.* The default *record size* of each file is 4096 words. The last record may be shorter but will be a multiple of 256 words. The beginning of each file contains the *file label* known to the file system.

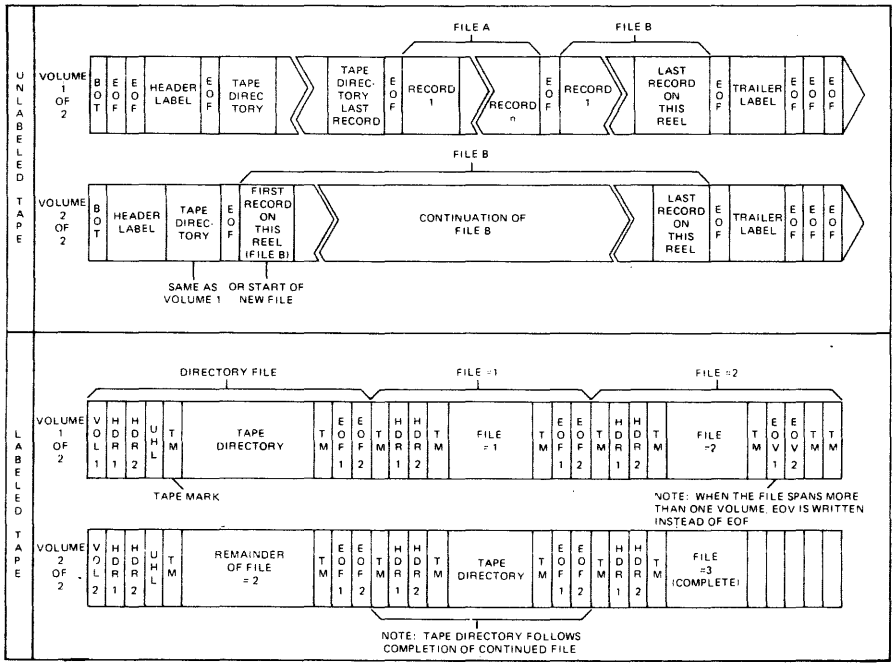


Figure 6-6. :STORE Tape Formats

DUMPING FILES OFF LINE

Any user with standard MPE capability can obtain a back-up copy of a particular user disc file or fileset, by copying fileset off-line to a serial storage device by issuing the :STORE command. The files are copied in a special format along with all descriptive information (such as *accountname,groupname*, and *lockword*), permitting them to be read back into the system later (by the :RESTORE COMMAND).

The :STORE and :RESTORE commands are used primarily as a back-up for files. They can be used to interchange files between installations if the accounts, groups, and creators of the files to be restored are defined in the destination system. Furthermore, if no destination device is specified in the :RESTORE command, MPE does not guarantee which devices will actually receive the files — if a device of the same type as the original device with sufficient storage space cannot be found, the file is restored to any device that is a member of the device class DISC.

Files currently open for output, input/output, update, or append access cannot be acted upon by a :STORE command. Files currently being stored or restored cannot be acted upon by a :STORE command. However, files loaded into memory (currently running programs) and files open for input only can be stored, since their contents cannot be altered.

While a file is being dumped, it is locked by MPE so that it cannot be altered or deleted until safely copied to tape or serial disc. If the job performing the :STORE/:RESTORE function is aborted by you or the Console Operator, those files not yet stored or restored will be *unlocked* during the processing of the abort.

The flow chart in figure 6-7 shows the checks performed against a file to ensure its eligibility for dumping.

After the tape is written, data showing the results of the :STORE command is printed. By default, this output is sent to the standard list device ($STDLIST). However, you can override this default and transmit the output to another file by issuing a :FILE command equating SYSLIST (the formal file designator by which the :STORE command executor references this list file) to another file. For example, a user at a terminal might transmit this output to a line printer by entering.

```
:FILE SYSLIST=MYFILE; DEV=LP
```

(Assume the device class LP is a high speed line printer.)

If the SHOW parameter is omitted from the :STORE command, only the total number of files actually stored, a list of files not stored, and a count of files not stored, are printed. But if SHOW is included, the listing of files appears, in the format shown in figure 6-8. An example printout is shown in the lower portion of figure 6-8.

In the format shown in figure 6-8, *xxx* is a value denoting the total number of files dumped onto tape; *yyy* denotes the number of files requested that were not dumped. The notations *filename, groupname,* and *acctname* under the FILES STORED heading name the individual files dumped, and their groups and accounts, respectively. The notation *ldn* indicates the logical device number (in decimal) of the device on which the file resides, and *addr* is the absolute address (in octal) of the file label. The notations *filename, groupname,* and *acctname* under the FILES NOT STORED heading, indicate the individual files not dumped, and their groups and accounts.

The notation *fileset#* shows the number of the fileset to which the particular file belongs (relative to its position in the *filesetlist* parameter). The notation *msg* is a message denoting the reason that the file was not dumped. These errors do not abort the file storing operation, which continues. Table 6-5 lists the messages and their meanings.

The following catastrophic errors abort the :STORE command:

Command syntax error.

File directory error.

File system error on the tape file (TAPE), list file (LIST), or temporary disc files (GOOD, ERROR) used by the :STORE command executor.

If :STORE encounters a disc error while storing a file, it will skip the rest of that file and send a warning message to the user; the entire :STORE operation will not be aborted. Files that are skipped in this manner cannot be restored with the :RESTORE command.

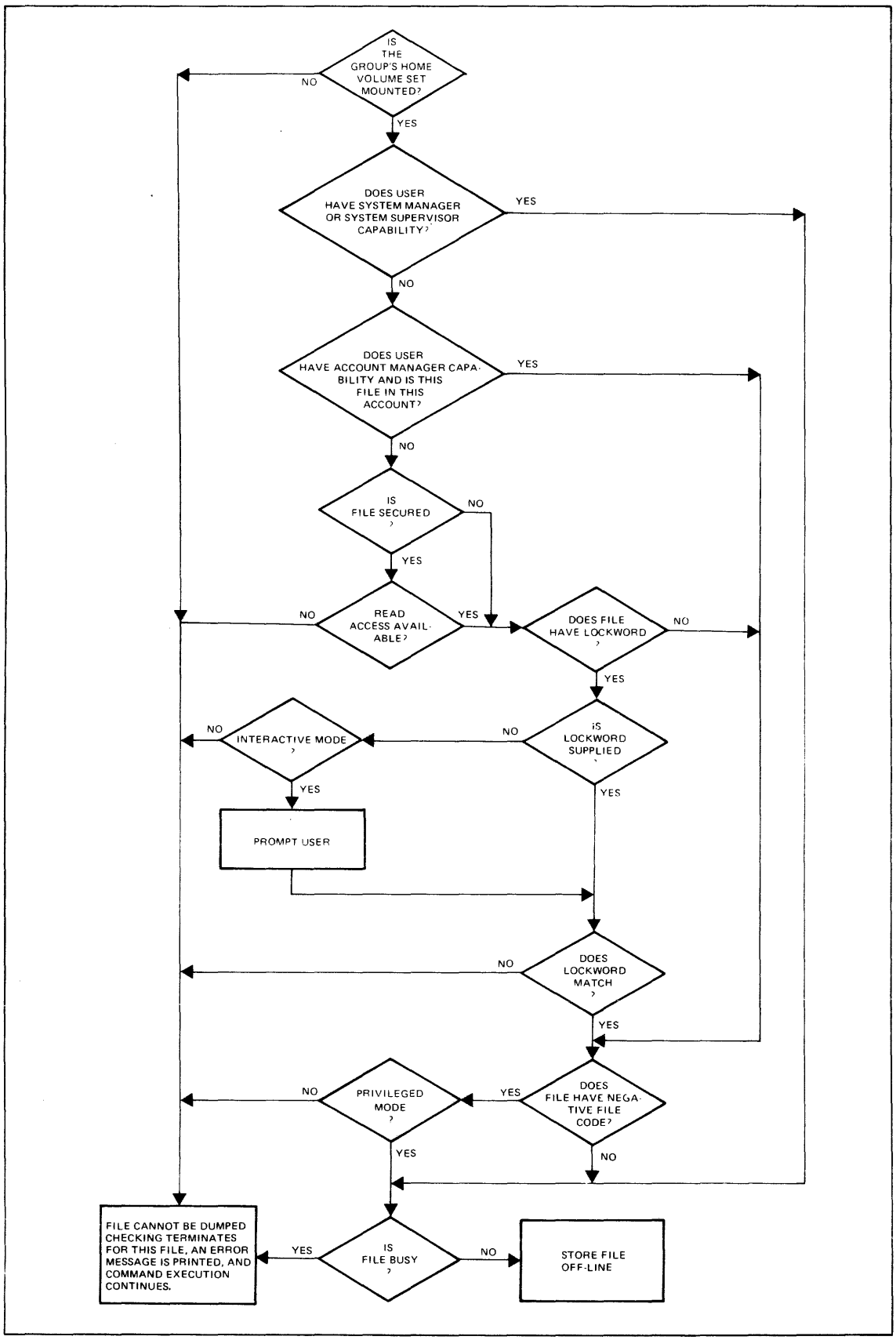


Figure 6-7. Checks for File Dump Eligibility

FILES STORED = xxx

FILE	*.GROUP*	*.ACCOUNT*	*LDN*	*ADDRESS*
filename1	*.groupname1*	*.acctname1*	*ldn1*	*addr1*
filename2	*.groupname2*	*.acctname2*	*ldn2*	*addr2*
.				
.				
.				
filenamen	*.groupnamen*	*.acctnamen*	*ldnn*	*addrn*

FILES NOT STORED = yyy

FILE	*.GROUP*	*.ACCOUNT*	*FILESET*	*CONDITION*
filename1	*.groupname1*	*.acctname1*	*fileset#*	*msg*
filename2	*.groupname2*	*.acctname2*	*fileset#*	*msg*
.				
.				
.				
filenamen	*.groupnamen*	*.acctnamen*	*fileset#*	*msg*

EXAMPLE

```
    FILES STORED = 13

      FILE      .GROUP    .ACCOUNT    LDN    ADDRESS

      DATA      .PUB      .SUPPORT    1      %24324
      FSMT      .PUB      .SUPPORT    1      %111052
      FSMTS     .PUB      .SUPPORT    1      %110775
      FTEST     .PUB      .SUPPORT    1      %111237
      FTESTJ1   .PUB      .SUPPORT    1      %24560
      FTESTJ2   .PUB      .SUPPORT    1      %24642
      FTESTJ3   .PUB      .SUPPORT    1      %25542
      FTESTJOB  .PUB      .SUPPORT    1      %23463
      FTESTJX   .PUB      .SUPPORT    1      %41207
      FTESTS    .PUB      .SUPPORT    1      %23603
      JUNKJOB   .PUB      .SUPPORT    1      %23533
      PEOF      .PUB      .SUPPORT    1      %374040
      PEOFS     .PUB      .SUPPORT    1      %373737

    FILES NOT STORED = 1

      FILE      .GROUP    .ACCOUNT FILESET   REASON

      K2861445.PUB        .SUPPORT    1   BUSY
```

Figure 6-8. List Output Format of :STORE Command

Table 6-5. :STORE Command Error Messages

MESSAGE	MEANING
ACCOUNT NOT IN DIRECTORY	Specified account does not exist.
GROUP NOT IN DIRECTORY	Specified group does not exist.
FILE NOT IN DIRECTORY	Specified file does not exist.
FILE BUSY	File is open for output, or is currently being stored or restored.
FILE CODE <0 AND YOU DO NOT HAVE PM CAPABILITY	A user without Privileged Mode capability is attempting to STORE a file with a negative file code.
LOCKWORD WRONG	The file lockword either was not provided or was specfied incorrectly.
READ ACCESS FAILURE	The user does not have read access to the specified file.
FILE LABEL ERROR	Due to a problem beyond the user's control, the file label is not valid.
GROUP HOME VOLUME SET NOT MOUNTED	The home volume set for the specified group has not been mounted.
IRRECOVERABLE TAPE ERROR	A tape error has occurred which cannot be corrected.
IRRECOVERABLE SERIAL DISC ERROR	An error has occurred on the serial disc that cannot be corrected.

RETRIEVING DUMPED FILES

You can read back into the system, onto disc, any file, or set of files, that has been stored off-line (on tape) by :STORE. The files referenced are attached to the appropriate groups and accounts, with previous account and group names, and lockwords all re-instated. File retrieval is requested with the :RESTORE command. *This command does not create any new accounts or groups. Any tape file to be restored will only be restored if the account name, group name and creator name exist on disc (in the system directory).* See the Initiator discussion of RELOAD and the ACCOUNTS option in particular.

If a copy of a file to be restored already exists on disc, you must have write access to the disc file (since it will be purged by :RESTORE). If this disc copy has a negative file code, you must have System Manager, System Supervisor or Privileged Mode capability to restore it.

Files *currently* open, loaded into memory, or being stored or restored, cannot be acted upon by a :RESTORE command.

The :RESTORE command performs the same checking performed by the :STORE command, to ensure a file's eligibility for retrieval. If the SHOW parameter is included in the :RESTORE command, a listing is produced showing which files were restored. Otherwise, a count of files restored, a list of files not restored, and a count of files not restored, are supplied.

As with the listing produced by :STORE, the listing output by :RESTORE is transmitted to a file whose formal designator is SYSLIST; if you do not specify otherwise, this file is equated, by default, to the standard list device ($STDLIST). An example of a typical :RESTORE with SHOW and KEEP options is shown in figure 6-9.

```
FILES RESTORED = 6

  FILE        .GROUP    .ACCOUNT    LDN    ADDRESS

  DATA        .PUB      .SUPPORT    1      %23463
  FTEST       .PUB      .SUPPORT    1      %24324
  FTESTJ1     .PUB      .SUPPORT    1      %23741
  FTESTJOB    .PUB      .SUPPORT    1      %23753
  FTESTS      .PUB      .SUPPORT    1      %23765
  PEOF        .PUB      .SUPPORT    1      %24354

FILES NOT RESTORED = 7

  FILE        .GROUP    .ACCOUNT FILESET      REASON

  FSMT        .PUB      .SUPPORT    1      ALREADY EXISTS
  FSMTS       .PUB      .SUPPORT    1      ALREADY EXISTS
  FTESTJ2     .PUB      .SUPPORT    1      ALREADY EXISTS
  FTESTJ3     .PUB      .SUPPORT    1      ALREADY EXISTS
  FTESTJX     .PUB      .SUPPORT    1      ALREADY EXISTS
  JUNKJOB     .PUB      .SUPPORT    1      ALREADY EXISTS
  PEOFS       .PUB      .SUPPORT    1      ALREADY EXISTS
```

Figure 6-9. List Output of :RESTORE with SHOW and KEEP

This format is exactly the same as that for the :STORE example of figure 6-8. The notation *msg* is an error message denoting the reason that the file was not restored. (These errors do not abort the file-restoring operation.) Table 6-5 lists the messages and their meanings.

The following catastrophic errors abort the :RESTORE command:

Command syntax error.

Disc input/output error (in system).

File directory error.

File system error on the tape file (TAPE), list file (LIST), or any of the three temporary files (GOOD, ERROR, and CANDIDAT) used by the :RESTORE command executor.

Improper tape; the tape used for input was not written in :STORE/:RESTORE format.

No continuation reel; the computer operator could not find a continuation reel for a multi-reel tape set.

Device reference error; the specification for the *device* parameter is illegal, or the device requested is not available.

Table 6-6. :RESTORE Command Error Messages

MEANING	MESSAGE
ACCOUNT DIFFERENT FROM LOGON	The file's account name is different from the name of the user's log-on account. (Users do not have save-access to groups outside of their log-on accounts.)
ACCOUNT DISC SPACE EXCEEDED	The account's disc space limit would be exceeded by restoring this file.
ACCOUNT NOT IN DIRECTORY	The account specified does not exist in the system.
ALREADY EXISTS	A copy of the file specified already exists on disc, and KEEP was also specified. The file was not replaced.
BUSY	The disc file is open, loaded, or being stored or restored at present.
CATASTROPHIC ERROR	A catastrophic error occurred while the system was restoring either this file or one previous to it on the tape, and the :RESTORE command was aborted. (Examples of such catastrophic errors are listed below.)
CREATOR NOT IN DIRECTORY	The creator of the file is not defined in the system.
DISC FILE CODE <0 AND NO PRIV MODE	One of the files (on disc) to be replaced has a negative file code, and the user does not have Privileged Mode capability.
DISC FILE LOCKWORD WRONG	The disc file has a lockword that does not match the lockword for the file on tape.
GROUP DISC SPACE EXCEEDED	The group's disc space limit would be exceeded by restoring this file.
GROUP NOT IN DIRECTORY	The group specified does not exist in the system.
NOT ON TAPE	The file specified is not on the tape.
OUT OF DISC SPACE	There is insufficient disc space to restore this file.
SAVE ACCESS FAILURE	The user does not have save-access to the group to which the file belongs.
TAPE FILE CODE <0 AND NO PRIV MODE	One of the files (on tape) to be restored has a negative file code, and the user does not have Privileged Mode capability.
TAPE FILE LOCKWORD WRONG	The tape file has a lockword that was not supplied by the user, or was specified incorrectly.
WRITE ACCESS FAILURE	The user does not have write-access to the copy of the file on disc.
FILE LABEL ERROR	The file is on disc and has a bad label.

Files created by the :STORE command and :SYSDUMP command are compatible. Thus, a file dumped through :SYSDUMP can be used as input for the :RESTORE command. However, :STORE/:RESTORE tapes or serial discs cannot be used as the first reel of system initiation tapes or serial discs since the operating system has not been copied to the tape.

PERMANENTLY ALLOCATING PROGRAMS AND PROCEDURES

A user with System Supervisor capability can enhance system efficiency by permanently allocating program files (sets of one or more segments) or procedures (individual segments). This is, in effect, a permanent allocation of the programs or procedures so that they are always in a loaded state with external references resolved regardless of whether they are being referenced. Since allocation of a code segment always results in the allocation of external segments referenced by it, when a program or procedure is permanently allocated, all of its external routines are also permanently allocated.

By permanently allocating large, frequently-used routines, a System Supervisor user can greatly reduce the system overhead that would otherwise be used in continually loading the routines. Additionally, permanent allocation reduces the number of segment linkage operations required.

The number of segment linkages is kept in a core resident Code Segment Table. (CST). In order to keep the table entries to a minimum, it is suggested that only programs or procedures which will be used by a large number of user programs be permanently allocated.

To permanently allocate a program file or procedure the :ALLOCATE command is used.

To deallocate a program file or procedure segment that has been permanently allocated, a System Supervisor user issues the :DEALLOCATE command. The program file or procedure segment is de-allocated immediately if it is not being used by a process; otherwise, it is deallocated as soon as it is no longer in use.

CODE SEGMENT TABLE

The Code Segment Table (CST) contains a list of code segments that have been allocated by the system (note that this also includes system code segments). The length of the CST is determined at system generation time, and consists of a number of 4-word segment descriptor entries. The actual number of entries in use at any one time is a variable, limited only by the length of the table. Entries are dynamically allocated by the operating system as programs are loaded and unloaded. Each entry contains control information about the segment, and gives its length and starting address.

The format of the CST is shown in figure 6-11. The first %300 entries are reserved for Segmented Libraries (i.e., those segments which may be accessed by any process within the limits of the Privileged User mode restrictions). The first entry (segment 0) of this portion of the CST is not used to point to a code segment. Instead, its first word contains the current number of sharable code segments ($\leqslant$191). Its remaining three words are reserved and may be used by system software. The first word of this entry (and hence the start of the CST) is pointed to by the Code Segment Table Pointer (CSTP) in memory address 0.

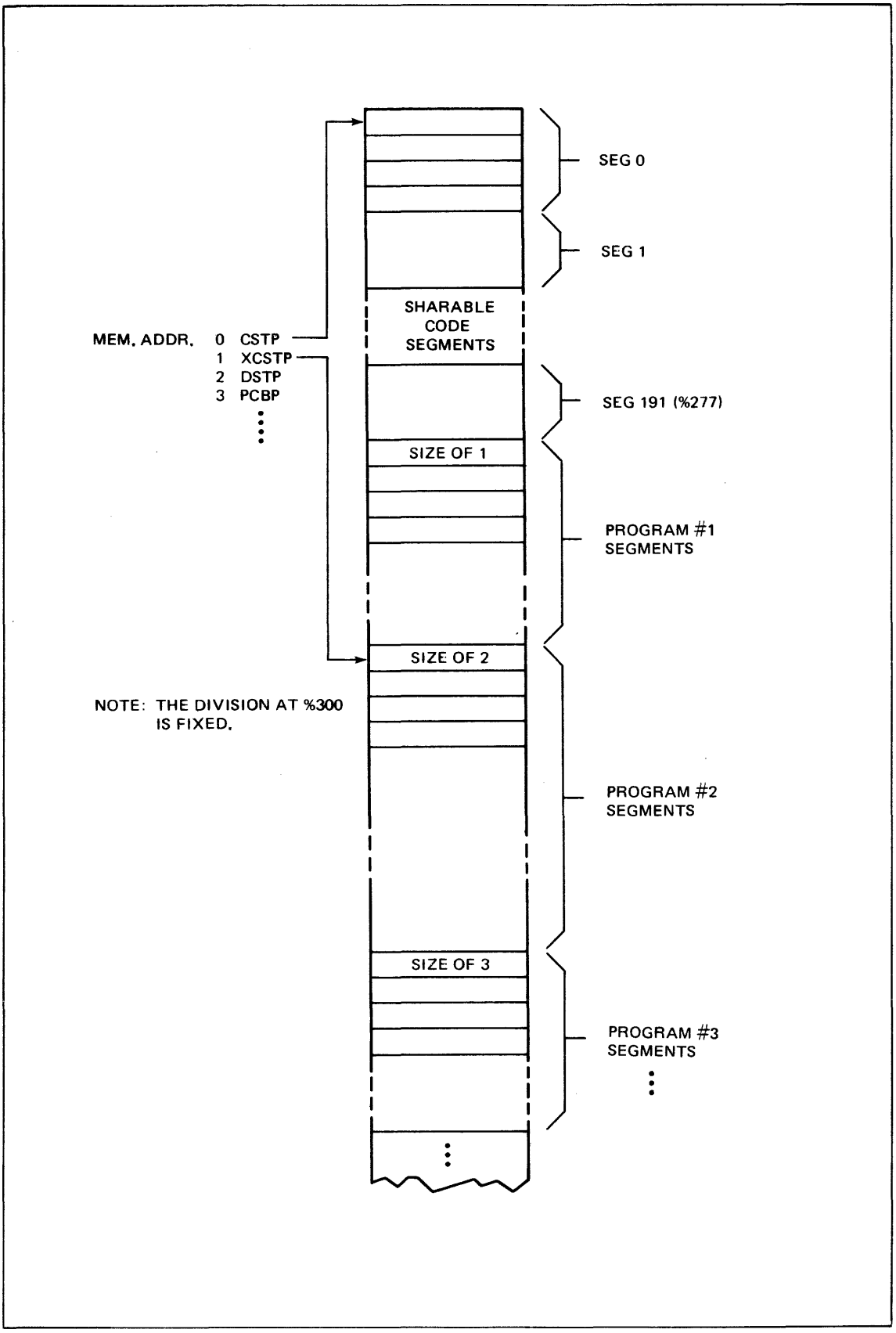
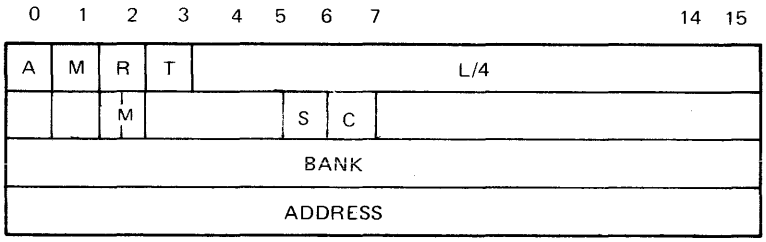


Figure 6-11. Code Segment Table Format

Segment 1 contains the routines needed to service internal interrupts. Segments 2 through n ($n \leq 191$) contain such things as service routines for external interrupts, system intrinsics, and library procedures.

The remainder of the table (beyond %300) consists of program segments. Each program may have up to 64 entries. Since the first entry must contain the number of segments in the program (identical to the segment 0 entry), it is actually limited to 63 segments. The Extended Code Segment Table Pointer (XCSTP) in memory address 1 is set by the operating system to point to the currently executing program (in figure 6-11, program number 2 is executing). Accessing the table is done via the PCAL, EXIT, IXIT, and DISP instructions and is completely transparent to the user. For example, if, on a PCAL, an external segment number lies in the range 0 < segment number ⩽ %277, the entry is accessed via the CSTP. If the segment number is in the range %300 < segment number ⩽ %377, then %300 is subtracted from the segment number and the entry is accessed via the XCSTP. This mechanism is built into the PCAL instruction.

The entries of the CST have the following form:



where

A = Absence bit, set to 1 if the code segment is absent from main memory.
M = Mode bit, set to 1 if the segment is to be executed in privileged mode.
R = Reference bit, set to 1 when this descriptor is accessed.
T = Trace bit, set to 1 if trace feature is used. This is checked by the PCAL instruction.
L = Length of the code segment.

NOTE

n is always an exact integer multiple of 4 and the maximum L = $2^{12} - 1$ = 4095. The maximum segment size, therefore, is 16,380 words.

IMI — Segment in motion into memory.
S — System segment.
C — Core resident.

BANK = Bank address, and points to the bank (if core resident) in which the segment resides.

ADDRESS = Absolute address of PB within BANK if the segment is present. Otherwise, the third and fourth words contain the absolute disc address.

JOB/SESSION PRIORITY REQUEST

A special capability granted to System Supervisors (and System Managers) is the use of the optional parameter HIPRI in the :HELLO and :JOB commands. (See the *MPE Commands Reference Manual.*) This is a request for maximum job-selection priority, and will cause the job to be dispatched regardless of the current job fence or execution limit.

LOGGING

MPE provides three separate logging facilities:

- System Logging
- Memory Error Logging
- User Logging

The operation of each facility is unique and has no effect on the other.

SYSTEM LOGGING

System Logging provides a record of certain resources by accounts, groups, and users on a job/session basis. System Logging can be activated with the Initiator (cold load) dialog or with the :SYSDUMP modification dialog.

While the Accounting System is managed by the System Manager and Account Managers, System Logging is managed by the System Supervisor. Like the Accounting System, System Logging can be used for billing or for simply obtaining an overview of system usage. Unlike the Accounting System, however, System Logging describes system usage on a job/session basis; in fact, it provides the only means for doing so, and for monitoring certain dynamic resources. It creates a running log of actual events, correlated with the job/session that caused each event. The events monitored are:

- Job/session initiation.
- Job/session termination.
- Process termination.
- File openings and closings.
- System shut-downs.
- Spooling.
- Console activities.
- MPE maintenance requests.
- Line closings and line disconnections.
- I/O errors.
- Private volume mounts and dismounts.
- Private volume set logical mounts and dismounts.
- Labeled tapes mounts.
- Stack underflow simulation.
- Program files.
- Diagnostic control unit activities, on Series 64 systems.

The running log is produced on one or more disc files known as *log files.* The events monitored are recorded on *log records* within such files, one logical record for each event. The System Supervisor can supply configuration information for log files, create, close, and purge such files, and display the amount of space used on the log file currently being written. However, the contents of log files — the individual log records — are not displayed nor otherwise used by MPE. Instead, various analysis routines must be written by the user and supplied for this purpose.

System Logging is activated during system configuration, through the Configurator/user dialogue (steps 108 through 114) discussed earlier. These steps allow you to:

- Enable or disable logging. (This option is also available during coldloading in step 92 of the Initiator-User dialogue discussed previously in this section.)
- Select the events to be logged.
- Determine the physical log-record size (in sectors).
- Determine the log-file size (in physical records).

When System Logging is enabled for the first time, the first log file is created and opened on disc, and records are written to it as the events to be logged occur. When this log file is full, it is closed by the system and a new log file is created and opened on disc. This new file becomes the *current* log file, and the previous log file is de-activated. Also, when a log file is closed by the System Supervisor, or when the system is shut-down and coldloaded again, the current log file is de-activated and a new log file is created and becomes the current log file. Thus, several log files can exist in the system at any time; one is the *current* log file, always an open file on disc, and the others are de-activated log files closed previously. The de-activated log files may exist on disc or may be copied to tape and optionally purged from disc, at the user's option. The implication here is that a log file becomes a permanent file after it is de-activated.

A log file is always designated by the name *LOGxxxx.PUB.SYS,* where *xxxx* is the log file number, ranging from 0000 to 9999; 0000 designates the first log file. With each new log file, *xxxx* is increased by 1.

Each time a new log file is created, the following message is output, showing the number (*x*) of the new (current) log file:

LOG FILE NUMBER x ON

GENERAL FORMAT. — All log files are initially created as files containing fixed-length records, but are later transformed into files formatted for variable-length records. They should always be treated as files containing variable-length records, accessed sequentially.

For a log file, the end-of-file pointer can point at the last record (block) written to the file (if the file is closed normally) or at any point beyond the last record written (if the system has not been shut down); in the latter case, all space following the last record is padded with zeros.

The general format of a log file is shown in figure 6-12.

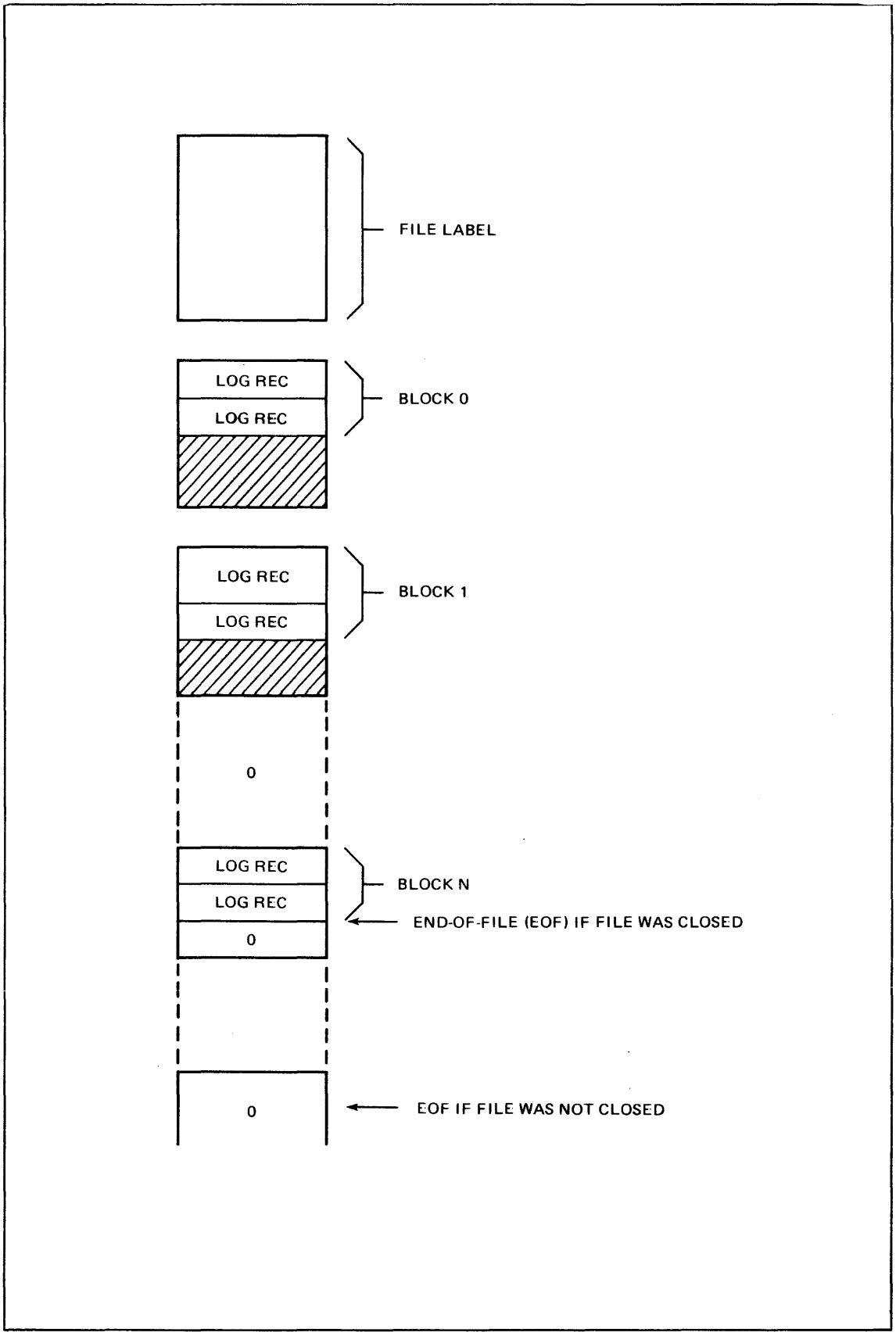


Figure 6-12. Log File Format

CONFIGURABLE CHARACTERISTICS. As noted above, you can specify, during system configuration, the physical log record (block) size and the total log file size. These characteristics are specified in steps 113 and 114, respectively, of the Configurator/User dialogue.

The block size is determined in disc sectors, and cannot exceed 1024 words. Since the Logging System uses an internal double-buffering system, block size is equivalent to the size of the log file buffer.

The file size is specified in number of blocks (physical records). The log file is always created with 16 disc extents; one extent is allocated initially, and additional extents are allocated one at a time, as needed. The number of sectors in each extent (except possibly the last) is determined by the following formula:

$$\text{Sectors} = \left[\left(\frac{\text{Log File Size}}{16}\right) \quad \text{x} \quad (\text{Log Record Size})\right]$$

FILE SECURITY. Because log files are files belonging to and created by System Logging, their creator, by implication, is the original System manager (MANAGER .SYS). A log file is created with the MPE default security provisions; since it is in the PUB .SYS group/account, this implies that it can be read by any user but can be changed only by Account Librarian (AL) users for SYS, and by group users (GU) for PUB .SYS. Once the log file is no longer current, access security of the file is changed by MPE from "any" to "creating user" only. Thus, the System Manager controls access to current and closed log files, but the System Supervisor manages the logging facility by specifying log file size, log record size and events logged, and by enabling/disabling logging.

LOG FILE COMMANDS. At any time, the System Supervisor can display the current log file's number and the amount (percentage) of available file space used. He does this by entering the :SHOWLOG command.

The System Supervisor can close the current log file and create a new log file by entering the :SWITCHLOG command.

FILE ERROR HANDLING. Two types of errors can prevent System Logging from maintaining the log file:

- Catastrophic Errors. These are physical input/output errors or unit failures. These errors are not recoverable; when they are detected, logging is disabled until the next cold-load.

- Managerial Errors. These are errors involving the creation and management of the log file through the MPE File Management System. These errors are sometimes recoverable if the System Manager or System Supervisor takes appropriate action. When such errors are detected, logging is temporarily suspended until recovery action appropriate to the error is completed and the System Supervisor enters the command :RESUMELOG.

  When logging resumes, a special log record is emitted that denotes the number of log events (and corresponding records) missed: total records missed, total job/session initiation records missed and total job/session termination records missed.

Log file errors are discussed further at the end of this section.

LOG RECORDS. Log records are written to the log file by MPE. This is done by calls to a special log intrinsic or process that requires the recording of a particular event. The log records can be subsequently accessed, manipulated, and displayed through user-supplied analysis routines.

Twenty-one types of log records can be entered; see table 6-7.

Table 6-7. Log Record Types

TYPE NO.	TYPE
0	Log Failure Record
1	System Up
2	Job Initiation Record
3	Job Termination Record
4	Process Termination Record
5	File Close Record
6	System Shut-Down Record
7	Power Failure Record
8	Spooler's Spoolfile Done Record
9	Line Disconnection
10	Line Close
11	I/O Error
12	Physical Mount/Dismount Record
13	Logical Mount/Dismount Record
14	Tape Labels Record
15	Console Log
16	Program File Event
17	Call Process Signals
18	DCE Provided Information
46	MPE Maintenance Request Log
47	Diagnostic Control Unit Log (for series 64 systems)

When logging is enabled, log failure and head records are always output. The remaining types of records may or may not be emitted, depending on the options selected during system configuration.

NOTE

Event 1, logging enabled, must be ON for any logging to take place. If event 2, job initiation, is ON and logging is enabled, the default input priority for jobs and sessions is 8. If event 8, spooler's spoolfile done record, is ON and logging is enabled, the default output priority is 8. Otherwise, the normal default for both input and output priorities is 13.

Events 1, 11, 46 and 47 will always be enabled. Requests to disable these types will be ignored.

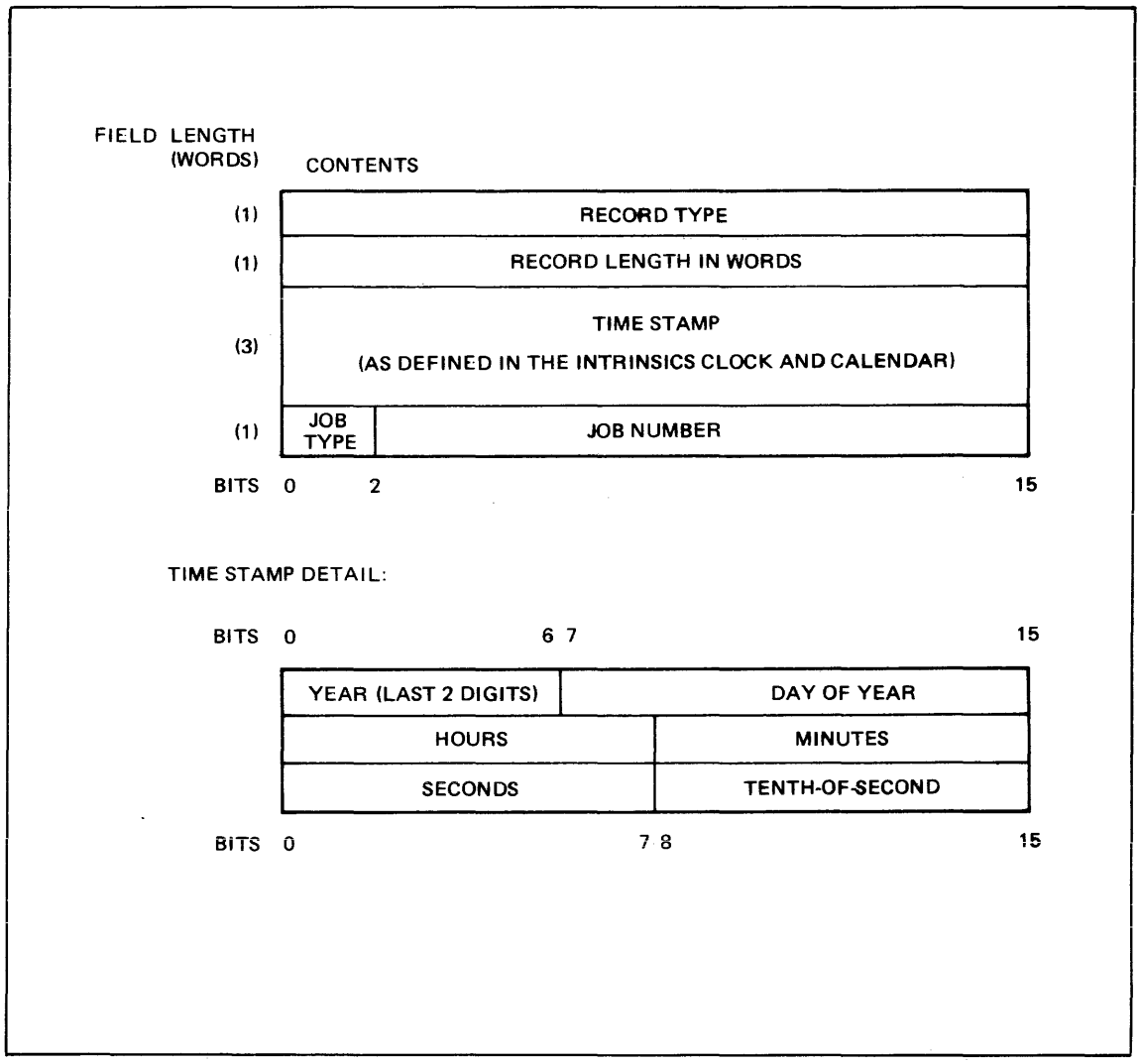


Figure 6-13. Log Record Heading Format

Log records, although different in format, length, and content, always have the same heading; see figure 6-13. In this format, the following definitions apply:

RECORD TYPE defines the record as one of the 17 types described above.

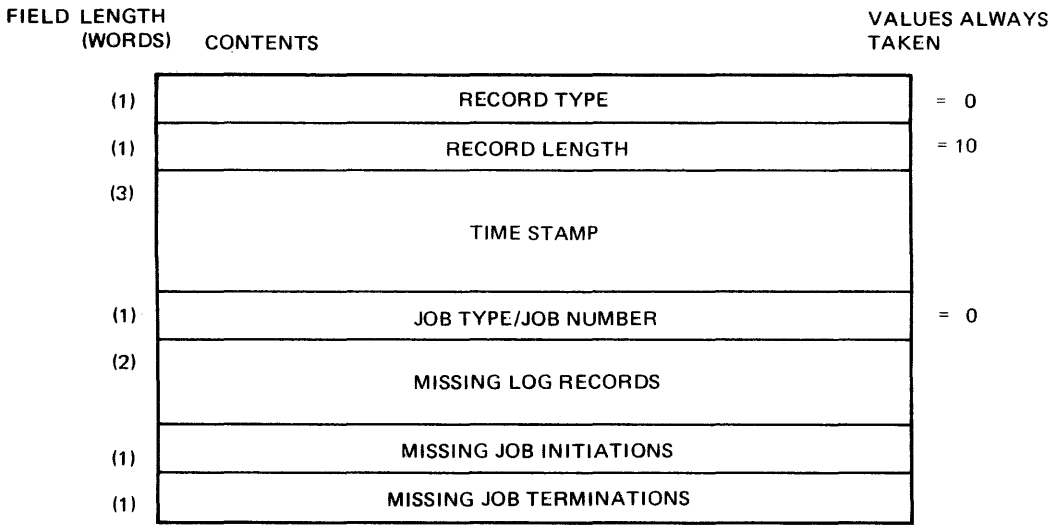
RECORD LENGTH defines the number of words that the record contains.

TIME STAMP is the date and time, in the format shown in the lower portion of figure 6-13.

JOB TYPE is the type of main process being run, where bits 0 and 1 have the following meaning:

(00)	System
(01)	Session
(10)	Job

JOB NUMBER is a number defining the job/session under which the log record has been output. If this last word is 0, this means that the record is related to the system and was not output for a user.



This record is issued after a recoverable logging error has occurred, and when logging resumes.

JOB TYPE/JOB NUMBER is always set to 0.

MISSING LOG RECORDS (2 words) is the total number of Log Record events occurring while the Logging System was suspended.

MISSING JOB INITIATIONS (1 word) is the number of jobs/sessions initiated while logging was suspended.

MISSING JOB TERMINATIONS (1 word) is the number of jobs/sessions terminated while logging was suspended.

Note: The last three numbers correspond to log records that would have been emitted if logging was not suspended.

Figure 6-14. Log Error Record (Type 0)

FIELD	LENGTH (WORDS)	CONTENTS	VALUES ALWAYS TAKEN
	(1)	RECORD TYPE	= 1
	(1)	RECORD LENGTH	= 17
	(3)	TIME STAMP	
	(1)	JOB TYPE/JOB NUMBER	= 0
SYSTEM NUMBER	(1)	UPDATE LEVEL	
	(1)	FIX LEVEL	
	(1)	CORE SIZE	
	(1)	AVAILABLE CST SIZE	
	(1)	AVAILABLE DST SIZE	
	(1)	AVAILABLE PCB SIZE	
	(1)	IOQ SIZE	
	(1)	TRL SIZE	
	(1)	ICS SIZE	
	(1)	(RESERVED)	
	(1)	MAX #OF CONCURRENT RUNNING JOBS/SESSIONS	

This record is issued after each cold-load or reload before the system is up.

JOB TYPE/JOB NUMBER is always set to 0.

SYSTEM NUMBER (2 words) is comprised of:

UPDATE LEVEL (1 word) is the update level of the system, composed of two ASCII characters.

FIX LEVEL (1 word) is the fix level of the system, composed of two ASCII characters.

CORE SIZE (1 word) is the main-memory size in K (1024) words of memory.

AVAILABLE CST SIZE (1 word) is the number of entries in the Code Segment Table.

AVAILABLE DST SIZE (1 word) is the number of entries in the Data Segment Table.

AVAILABLE PCB SIZE (1 word) is the number of entries in the Process Control Block.

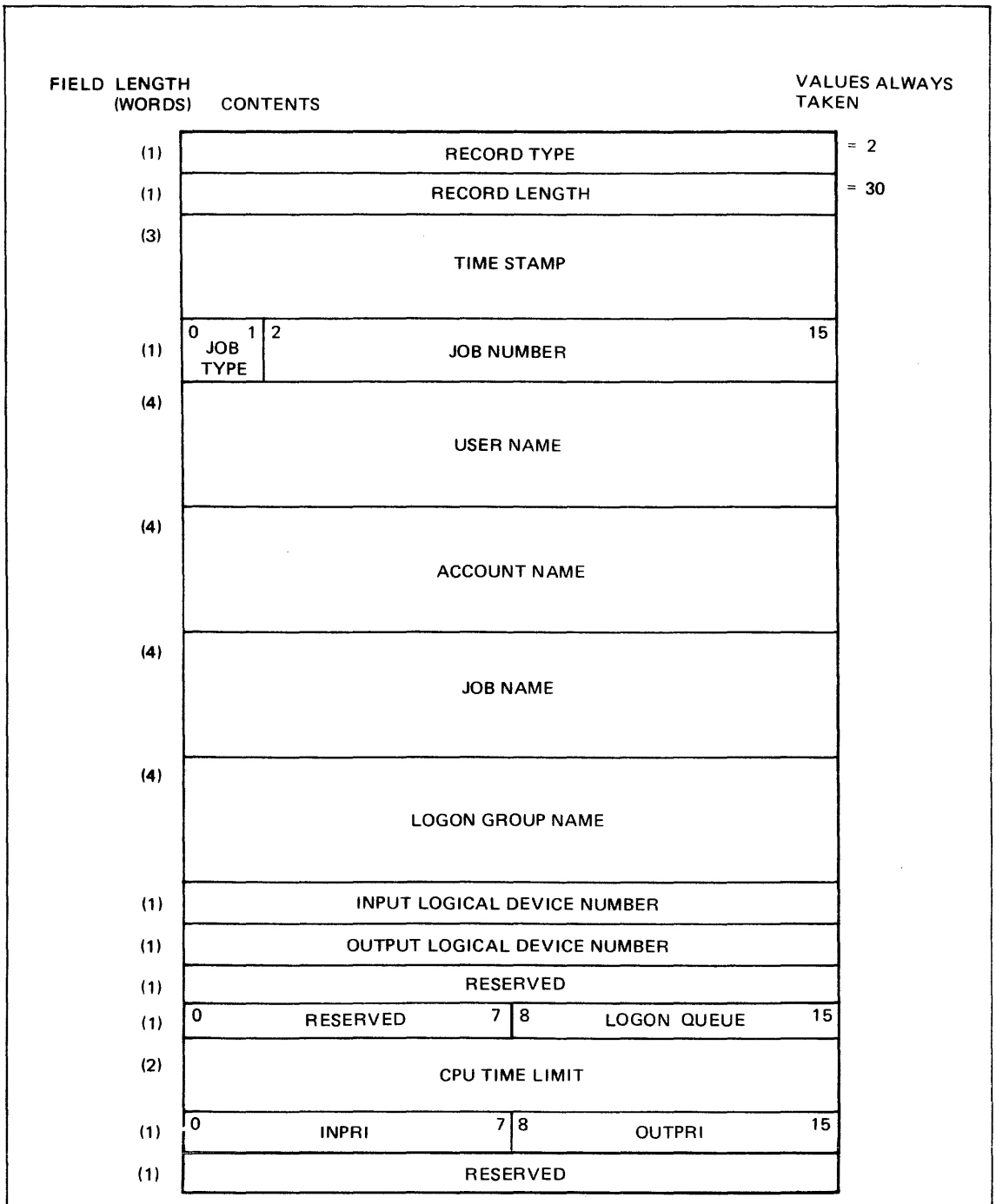
IOQ SIZE (1 word) is the number of entries in the Input/Output Queue.

TRL SIZE (1 word) is the number of entries in the Time Request List.

ICS SIZE (1 word) is the number of words in the Interrupt Control Stack.

MAX # OF RUNNING JOBS (1 word) is the maximum number of running jobs/sessions allowed in execution.

Figure 6-15. Head Record (Type 1)



This record is issued by the MPE Command Interpreter following a successful logon to MPE.

JOB TYPE/JOB NUMBER (1 word) are the type and number of job/session.

USER NAME (4-word array) is the user name as specified on the :JOB card or in the :HELLO command, left-justified, blank-padded.

ACCOUNT NAME (4-word array) is the account name as specified on the :JOB card or in the :HELLO command, left-justified, blank-padded.

Figure 6-16. Job Initiation Record (Type 2)

JOB NAME (4-word array) is the job name as specified on the :JOB card, or the session name as specified in the :HELLO command, left-justified, blank-padded. If *jobname* or *sessionname* was omitted, the array is filled with blanks.

LOGON GROUP NAME (4-word array) is the group name under which the log-on was performed, left-justified, blank-padded.

INPUT LOGICAL DEVICE NUMBER (1 word) is the logical device number of the standard input device for the job/session.

OUTPUT LOGICAL DEVICE NUMBER (1 word) is the logical device number of the standard listing device for the job/session.

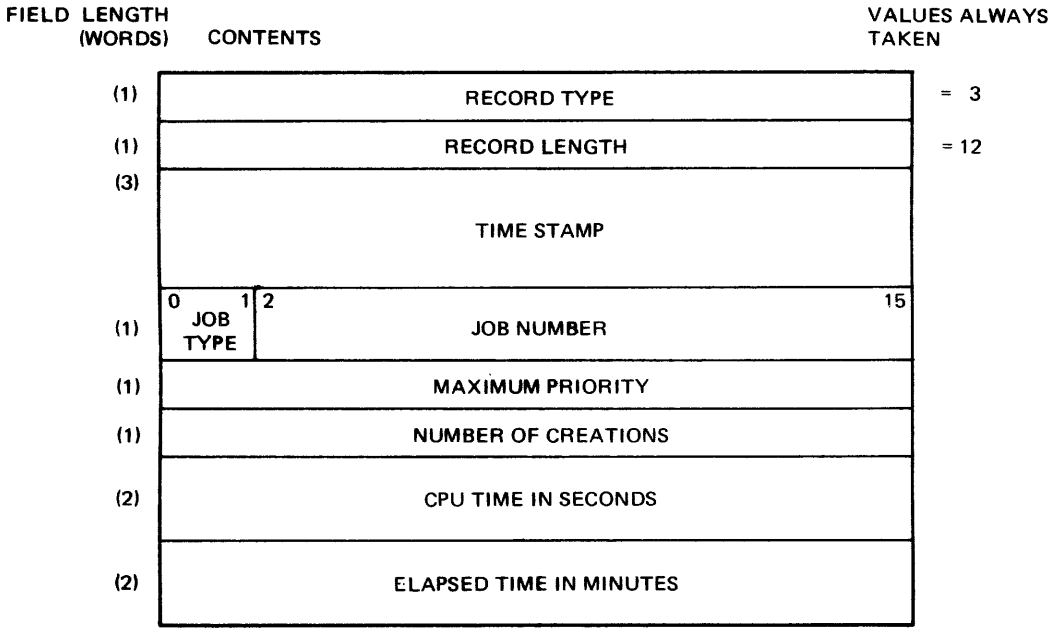
LOGON QUEUE (1 byte, right justified) is a single ASCII character representing the subqueue in which the job/session will execute; i.e. "B", "C", "D", or "E".

CPU TIME LIMIT (2 words) is a double-word showing the central processor time limit as specified on the :JOB card or in the :HELLO command. If no limit applied, the field contains -1. If omitted, the field contains 0.

INPRI (1 byte) is the job selection priority as defined on the :JOB card, or the default value if no priority specified.

OUTPRI (1 byte) is the job output priority as specified in the OUTCLASS parameters of the :JOB card, or 0 if no priority is specified or the job type is a session.

Figure 6-16. Job Initiation Record (Type 2) (Continued)



This record is issued at the end of a job or session, regardless of the cause of termination.

JOB TYPE/JOB NUMBER (1 word) are the type and number of the job/session.

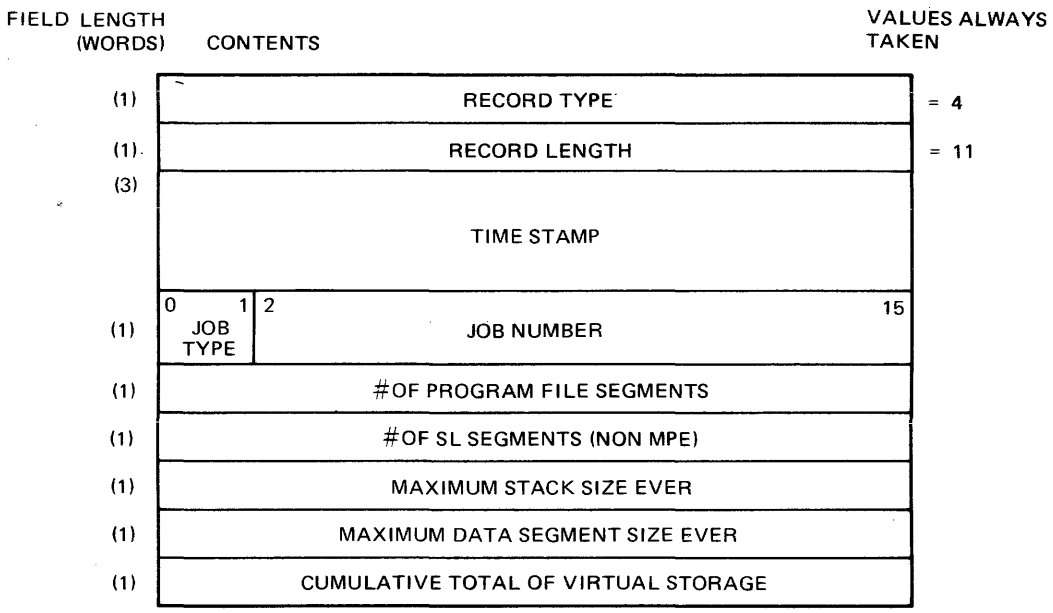
MAXIMUM PRIORITY (1 word) is the lowest priority number (highest priority) ever run under by any process of the job/session. This value can range from 0 to 255.

NUMBER OF PROCESSES CREATED (1 word) is the total number of processes created under the main process during the job/session.

CPU TIME (2 words) is a double-word containing the total central processor time used (in seconds) by all processes of the job/session.

ELAPSED TIME (2 words) is a double-word containing the wall clock time (in minutes) during which the job/session has existed in the system. For a session, this time is called *connect time.*

Figure 6-17. Job Termination Record (Type 3)



This record is issued when a user process other than a Main Process terminates.

JOB TYPE/JOB NUMBER (1 word) are the type and number of the job/session.

NUMBER OF PROGRAM FILE SEGMENTS (1 word) is the number of segments contained in the program file loaded on behalf of the process.

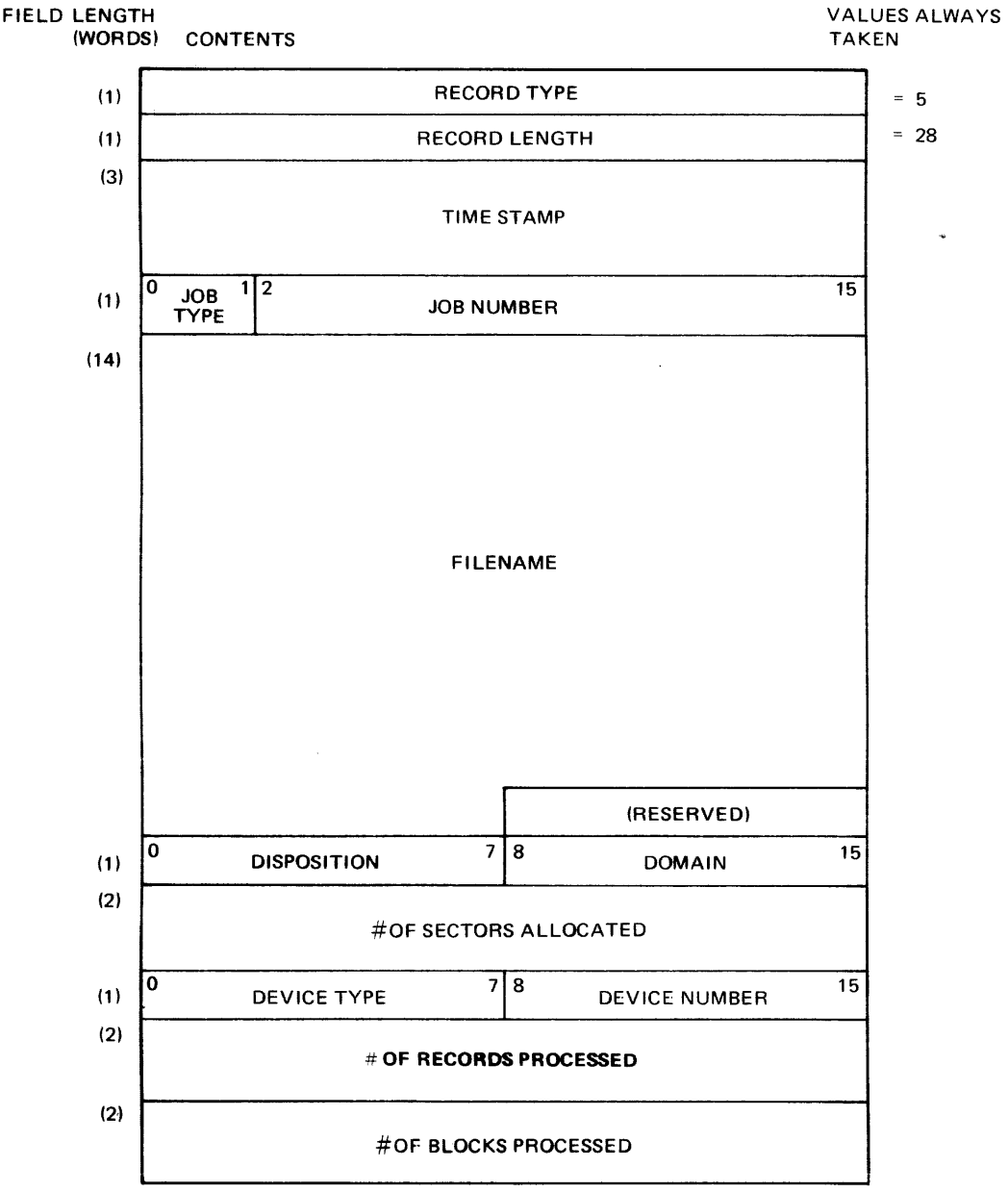
NUMBER OF NON-MPE SL SEGMENTS (1 word) is the number of segments from the segmented library (excluding MPE), loaded on behalf of the process.

MAXIMUM STACK SIZE EVER (1 word) is the largest size (in words) ever attained by the stack during process life.

MAXIMUM DATA SEGMENT SIZE EVER (1 word) is the largest size (in sectors) ever attained by an extra data segment during process life.

TOTAL AMOUNT OF VIRTUAL STORAGE REQUESTED (1 word) is the total amount of disc space (in sectors) requested for data (stack and extra data segments) during process life.

Figure 6-18. Process Termination Record (Type 4)



This record is issued whenever a file is closed.

JOB TYPE/JOB NUMBER (1 word) are the type and number of the job/session.

FULL FILENAME (27 bytes) contains the *filename.groupname.accountname* of the file; each name is eight bytes long, left justified with trailing blanks, with the 27th byte containing garbage.

DISPOSITION (1 byte) is the file disposition as specified in the FCLOSE intrinsic.

Figure 6-19. File Close Record (Type 5) (Sheet 1 of 2)

DOMAIN (1 byte) is the file domain, as specified in the FOPEN intrinsic.

NUMBER OF SECTORS ALLOCATED (2 words) is the physical space (in sectors) actually reserved on disc for the file. (When the file does not reside on disc, this value is 0.)

DEVICE TYPE (1 byte) is the device type of the device on which the file resides, as returned by the FGETINFO intrinsic.

DEVICE NUMBER (1 byte) is the logical device number of the device on which the file resides.

RECORDS PROCESSED (2 words) is the number of records processed since the last FOPEN on that file by the process.

BLOCKS PROCESSED (2 words) is the number of blocks written and read to and from the file since the last FOPEN on the file by the process.

Figure 6-19. File Close Record (Type 5) (Sheet 2 of 2)

FIELD LENGTH (WORDS)	CONTENTS	VALUES ALWAYS TAKEN
(1)	RECORD TYPE	= 6
(1)	RECORD LENGTH	= 9
(3)	TIME STAMP	
(1)	0 1 JOB TYPE / 2 15 JOB NUMBER	= 0
(1)	#OF JOBS	
(1)	#OF SESSIONS	
(1)	(RESERVED)	= 0

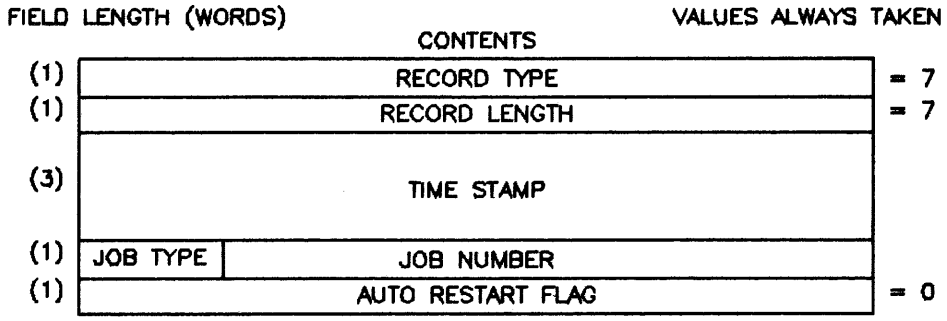
This record is output when the system is shut down (by the =SHUTDOWN console command. See the MPE Console Operator's Guide.).

JOB TYPE/JOB NUMBER is always set to 0.

NUMBER OF JOBS ON SYSTEM (1 word) is the number of jobs on the system when the command took effect.

NUMBER OF SESSIONS ON SYSTEM (1 word) is the number of sessions on the system when the command took effect.

Figure 6-20. Shut-Down Record (Type 6)



AUTO RESTART FLAG (1 word) is a logical value representing the state of the restart flag. This record is output when a restart occurs following a power failure.

Figure 6-21. Power Fail Record (Type 7)

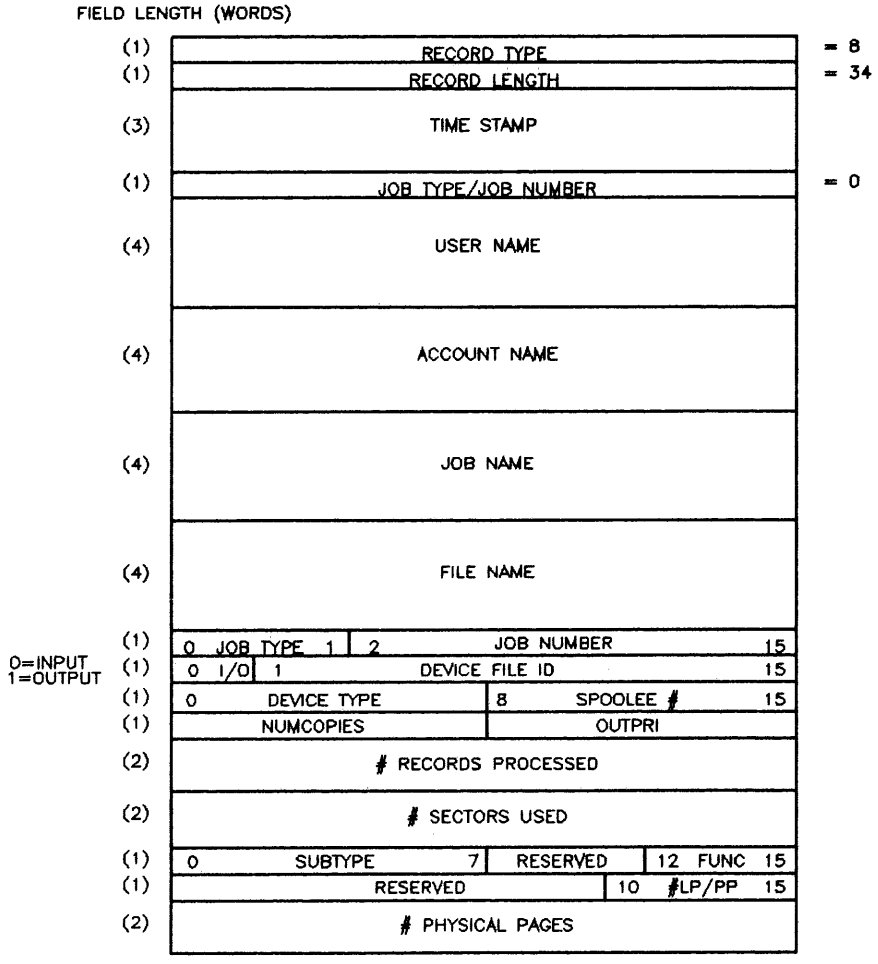


Figure 6-22. Spooler's Spoolfile Done Record (Type 8)

If the subtype is that of the 2680A, subtype 8, then the following two fields will be non-zero. For any other lineprinter, the fields will contain zeros.

PHYSICAL PAGES: The total number of physical pages printed.

LP/PP:

If the spoolfile is using an LPX type environment file, this field is defined as the number of logical pages per physical page when the spoolfile ended.

NOTE

This value can change while the spoolfile is printing.

FUNC (4 bits) is the last operation the spooler performs on the spoolfile and is defined as:

0 = normal completion
1 = delete spoolfile
2 = defer spoolfile
3 = relink spoolfile

This record is output when a spooler completes loading a spoolfile.

JOB TYPE is defined as:

(00) SESSION' = Spoolfile originally created on another system by a session.

(01) SESSION = Spoolfile created for a session on this system.

(10) JOB = Spoolfile created for a job on this system.

(11) JOB' = Spoolfile originally created on another system by a job.

For job types (00) and (11), SPOOK was used to create the spoolfile on this system. In all cases, the JOB NUMBER is the number of the job/session which originally created the spoolfile.

DEVICE TYPE is defined as:

Input - zero (0) for input spoolfiles

Output - 32 for output spoolfiles (device type of a lineprinter)

SPOOLEE # is defined as:

Input - The logical device number of the disc on which the spoolfile resided.

Output - The logical device number of the printer used.

NUMCOPIES:

This value represents the number of copies that are yet to be printed for this spoolfile. For multiple copy spoolfiles, this number is decremented after each copy has been printed and a different spooling logging record exists for each copy printed. If there is only one copy to be printed, the field is zero (0).

RECORDS PROCESSED is defined as:

Input - The number of lines in the input spoolfile.

Output - The number of lines printed for this spoolfile.

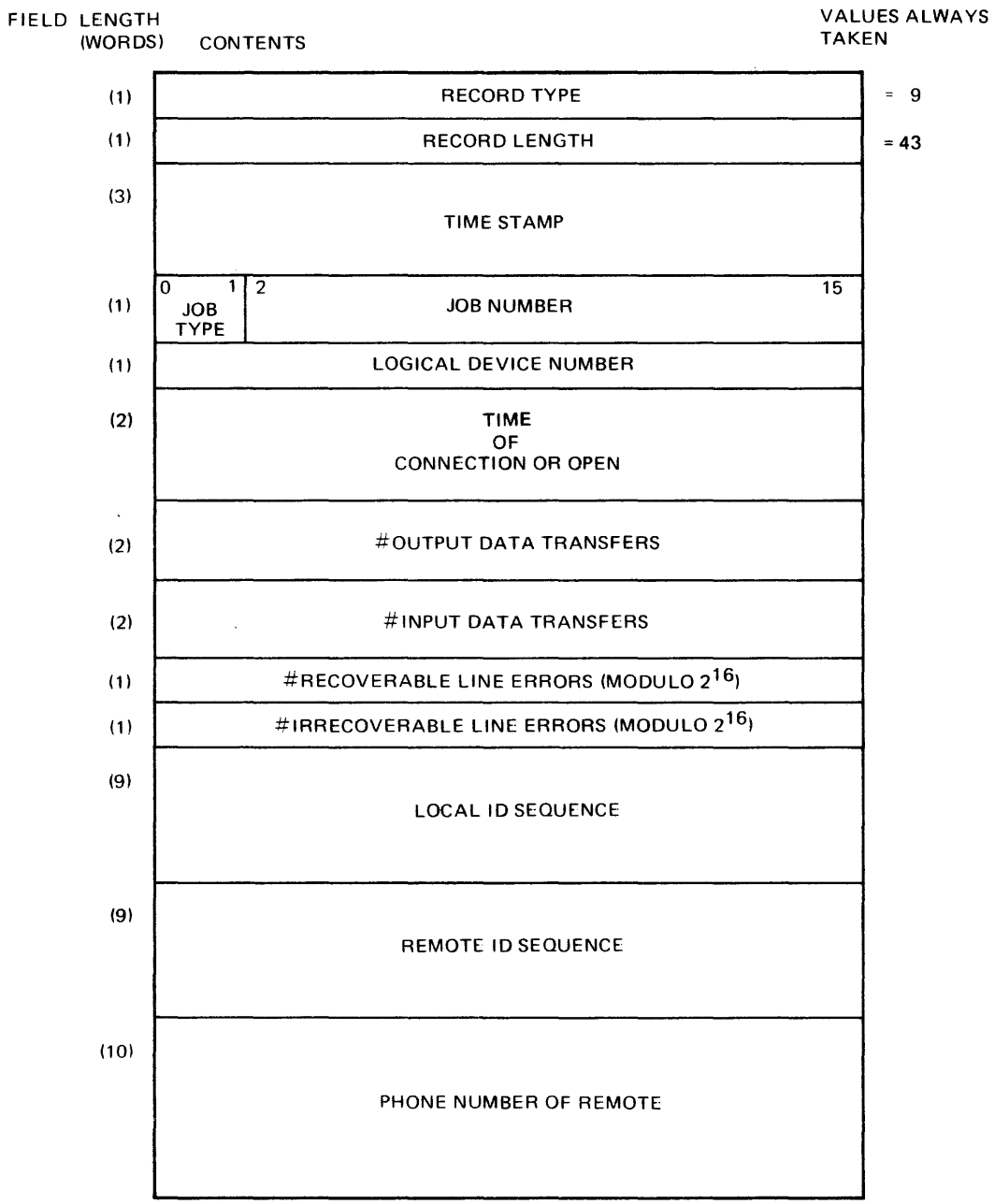
SECTORS USED is defined as:

The number of disc sectors that the spoolfile occupied on disc.

SUBTYPE is defined as:

Input - Zero (0) for input

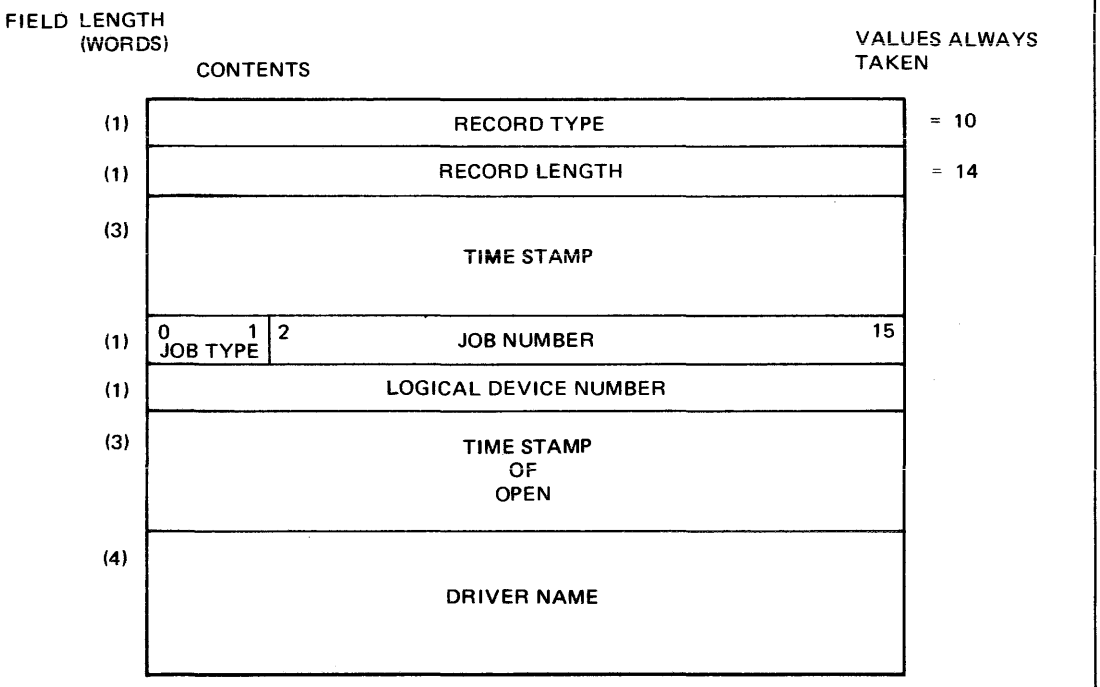
Output - The subtype of the lineprinter used.



ID sequences are significant for both switched and non-switched lines. The first byte is the length of the sequence. The remainder is the actual ID sequence (up to 16 bytes, left justified with trailing blanks).

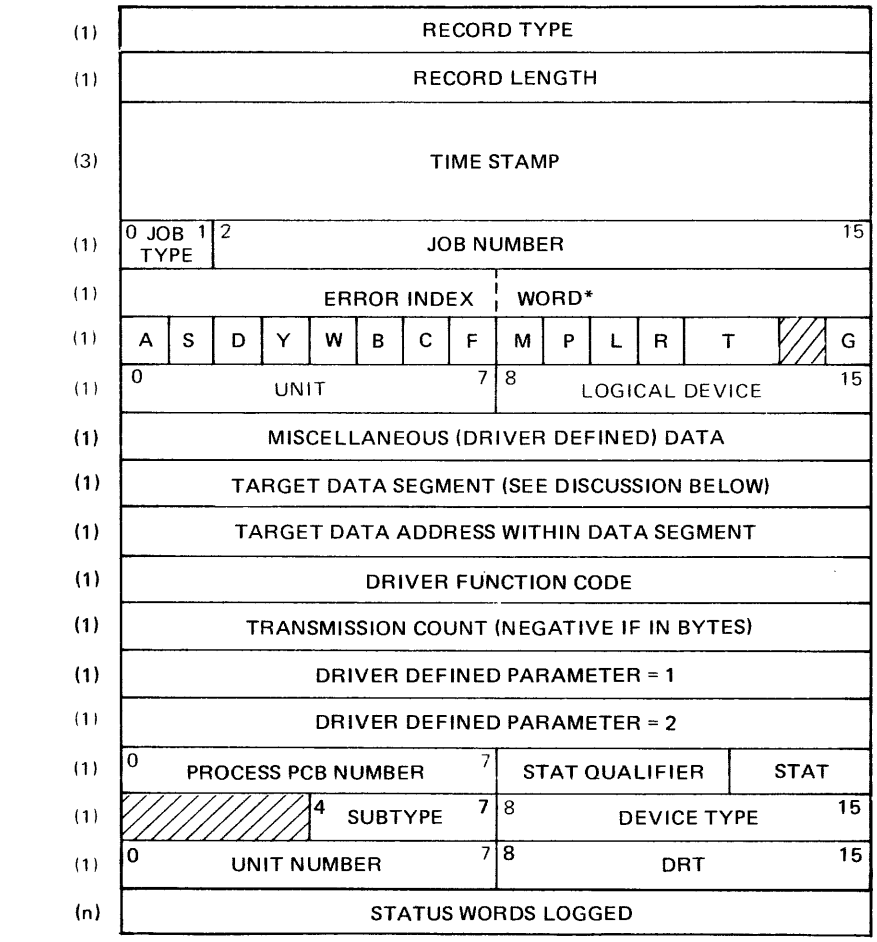
The phone number is significant only if the local system (MPE) performed the dialing for an outgoing call (up to 20 characters, left justified with trailing blanks).

Figure 6-23. Line Disconnection Record (Type 9)



The driver name may be up to eight ASCII characters in length, left justified with trailing blanks.

Figure 6-24. Line Close Record (Type 10)



The following is an explanation of the bits in the word following the error index word. A 1 in any bit position indicates:

A	(ABORT)	Request has been aborted externally.
S	(SPECIAL)	Special handling is to be applied to this request; for disc this indicates a Memory Management request.
D	(DIAG)	Diagnostic request (not used).
Y	(SYSBUFR)	Indicates the target data segment is irrelevant; the target data address is the absolute main memory address of the buffer.
W	(IOWAKE)	Wake caller on completion of request.
B	(BLOCKED)	Blocked I/O; caller is waited until request is completed.
C	(COMPLETED)	Request has been completed and caller is awake if he had so specified.
F	(DATAFRZN)	Data segment has been made present and is frozen
M	(MAMERRORD)	MAM error on data segment make-present.
P	(PREQ)	This request has been started but was preempted by a MAM request.
L	(SFAIL)	Start SIO failure in General Interrupt Processor.
R	(PFAIL)	The I/O has been aborted because of a power fail.
T	(PREEMPT)	Preemptive type code: 1=soft, 2=hard.
G	(MSGDONE)	A message request reply has completed.

* Error index word - 1st byte in No. of error status words logged. 2nd byte is Device Information Table relative index to words logged.

Figure 6-25. I/O Error Record (Type 11) (Sheet 1 of 2)

Driver Function Code is generally a number indicating a particular driver action, such as:
0=read, 1=write.

Transmission Count is the final transfer count (if any) for this I/O request. If positive, it indicates words; if negative, it indicates a byte count.

Driver Parameters 1 and 2 are defined within the I/O system for various drivers.

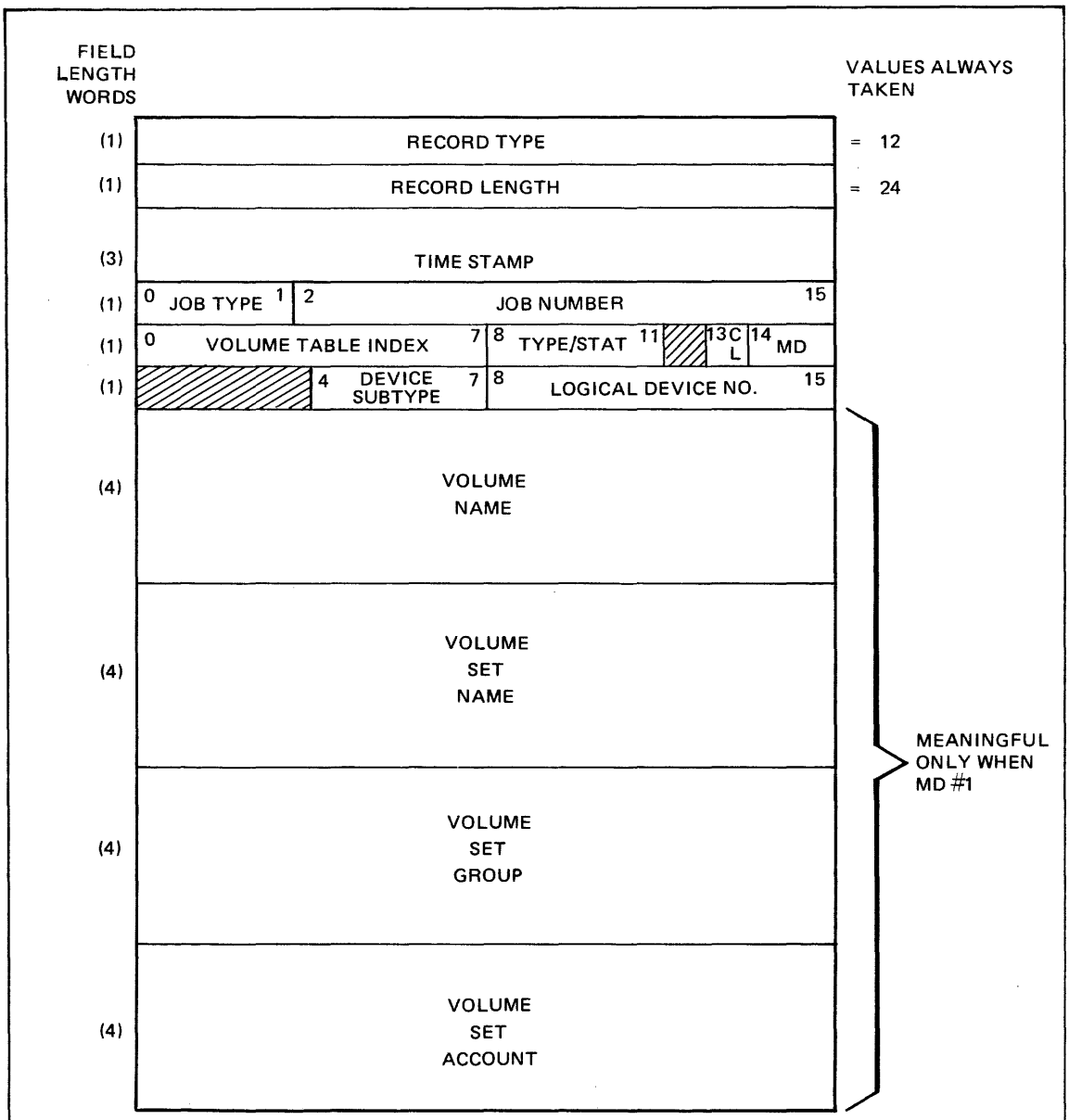
The Process PCB Number is zero if not associated with a user process.

STAT (general status) indicates current and resultant state of the request according to the following codes:

0	Not started, or awaiting completion.
1	Successful completion.
2	End of file detected.
3	Unusual condition (normally recoverable).
4	Irrecoverable error.

The Device Subtype and Type are as defined in Appendix A.

Figure 6-25. I/O Error Record (Type 11) (Sheet 2 of 2)



MD = 0 MOUNT
= 1 DISMOUNT
= 2 NEW VOLUME CREATED BY INIT FUNCTION

CL = 1 VOLUME MOUNTED AT COLDLOAD TIME
= 0 MOUNT/DISMOUNT DETECTED BY AUTO RECOGNITION.

TYPE/STAT — VOLUME TYPE /STATUS

= 0 VALID PRIVATE VOLUME
= 1 SCRATCH VOLUME
= 2 SYSTEM VOLUME
= 3 UNFORMATTED VOLUME
= 4 UNREADABLE VOLUME
= 5 SERIAL DISC VOLUME
= 6 FOREIGN VOLUME

Figure 6-26. Physical Mount/Dismount Record (Type 12)

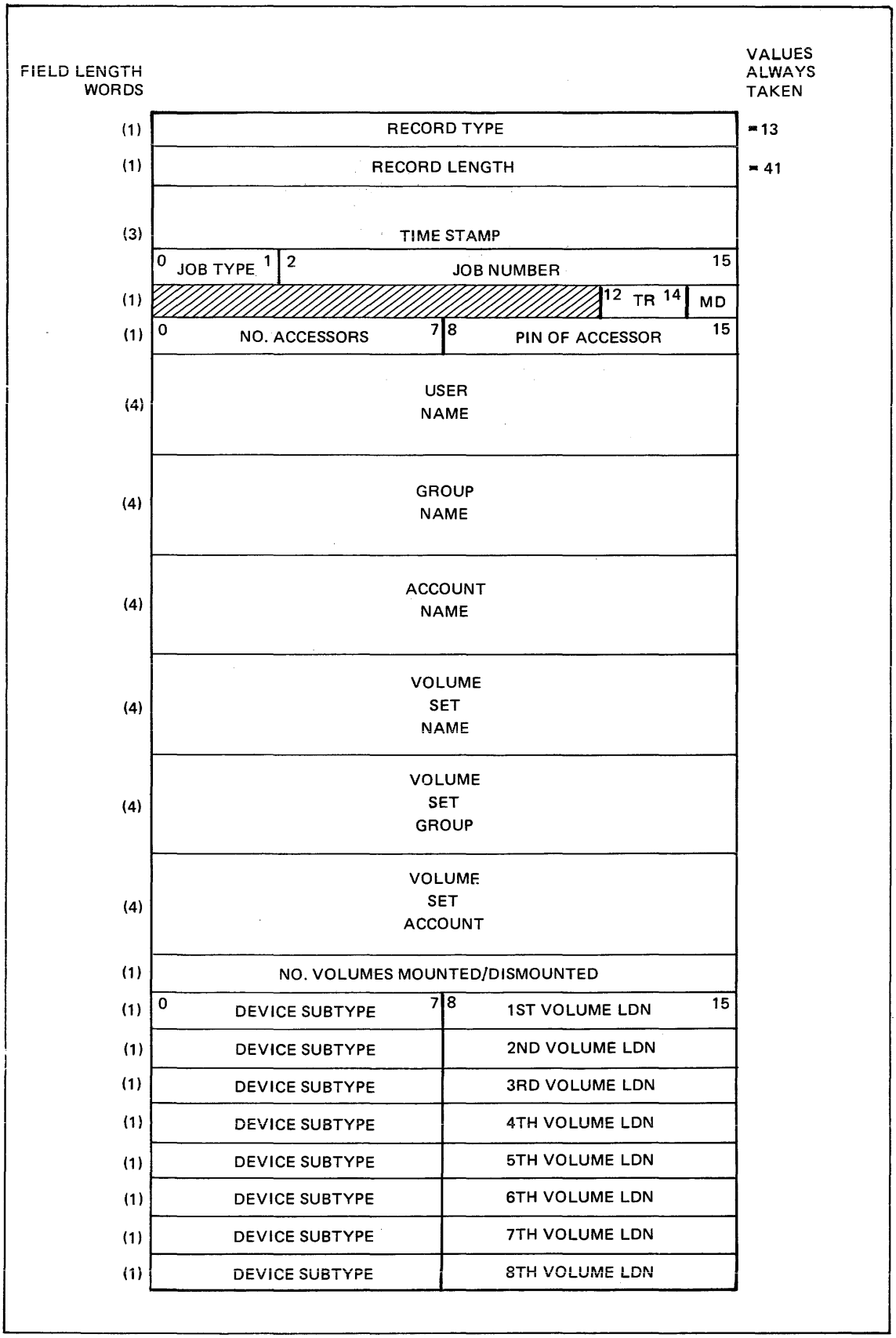


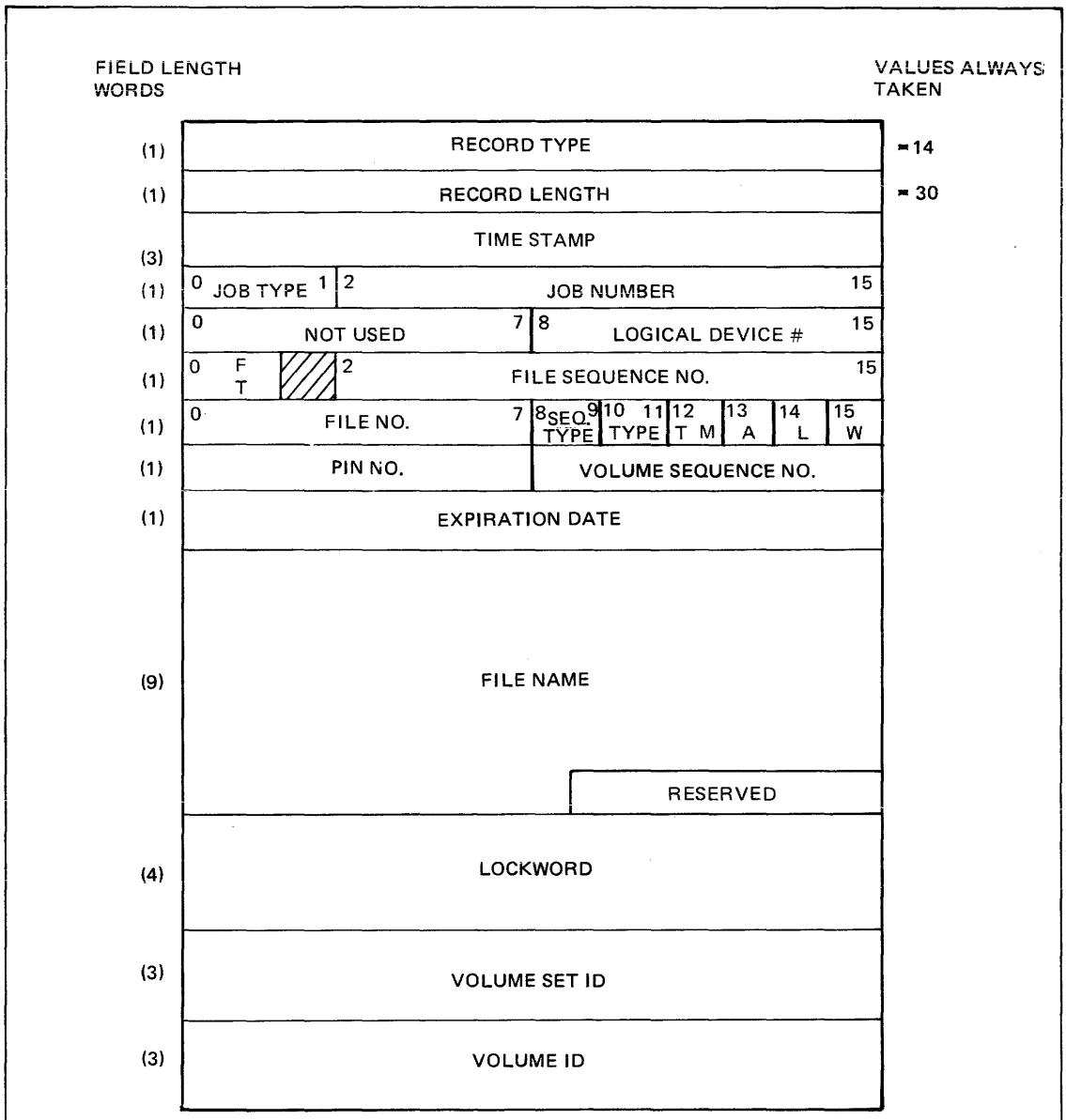
Figure 6-27. Logical Mount/Dismount Record (Type 13) (Sheet 1 of 2)

```
MD  = 0  MOUNT
    = 1  DISMOUNT

TR — TYPE OF REQUEST

    = 0  :MOUNT/:DISMOUNT WITH DIRECTORY BINDING
    = 1  :MOUNT/:DISMOUNT WITHOUT DIRECTORY BINDING
    = 2  UNCONDITIONAL IMPLICIT MOUNT
    = 3  CONDITIONAL IMPLICIT MOUNT
    = 4  =MOUNT/=DISMOUNT
    = 5  DISMOUNT DUE TO JOB/SESSION TERMINATION
```

Figure 6-27. Logical Mount/Dismount Record (Type 13) (Sheet 2 of 2)

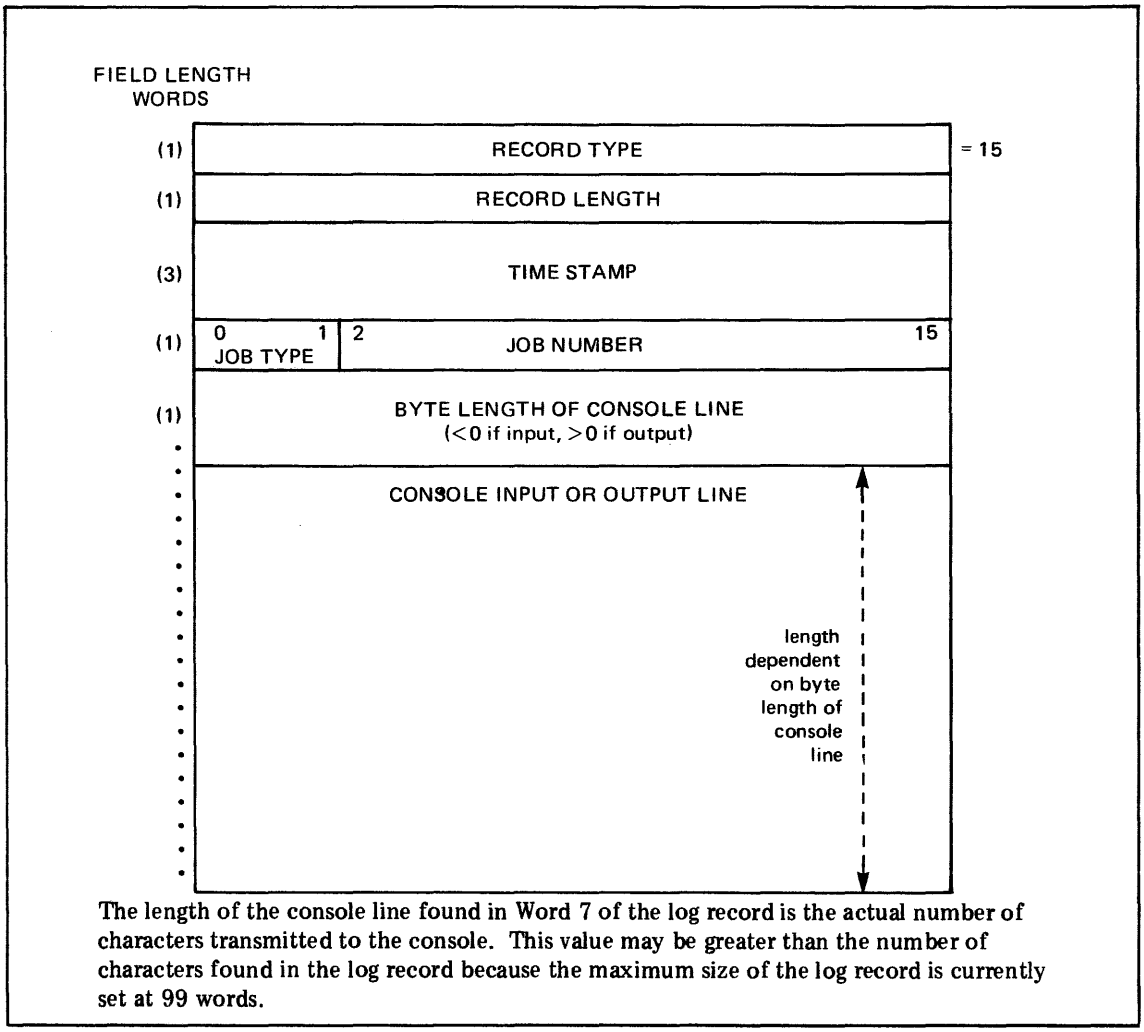


TYPE = 2 ANSI-STANDARD LABEL
TYPE = 3 IBM-STANDARD LABEL

SEQ. TYPE = 0 SEARCH FOR MATCH ON FILE NAME
= 1 NEXT OR DEFAULT
= 2 ADD FILE TO END OF VOLUME SET
= 3 SPECIFIED FILE SEQUENCE NUMBER

W = WAITING FOR MOUNT
L = JOB ENTRY LINKED TO LOGICAL DEVICE
A = USER HEADER LABEL
TM = USER TRAILER LABEL
FT = 1 AT LEAST ONE READ/WRITE COMPLETED

Figure 6-28. Tape Labels Log Record (Type 14)



The length of the console line found in Word 7 of the log record is the actual number of characters transmitted to the console. This value may be greater than the number of characters found in the log record because the maximum size of the log record is currently set at 99 words.

Figure 6-29. Console Log Record (Type 15)

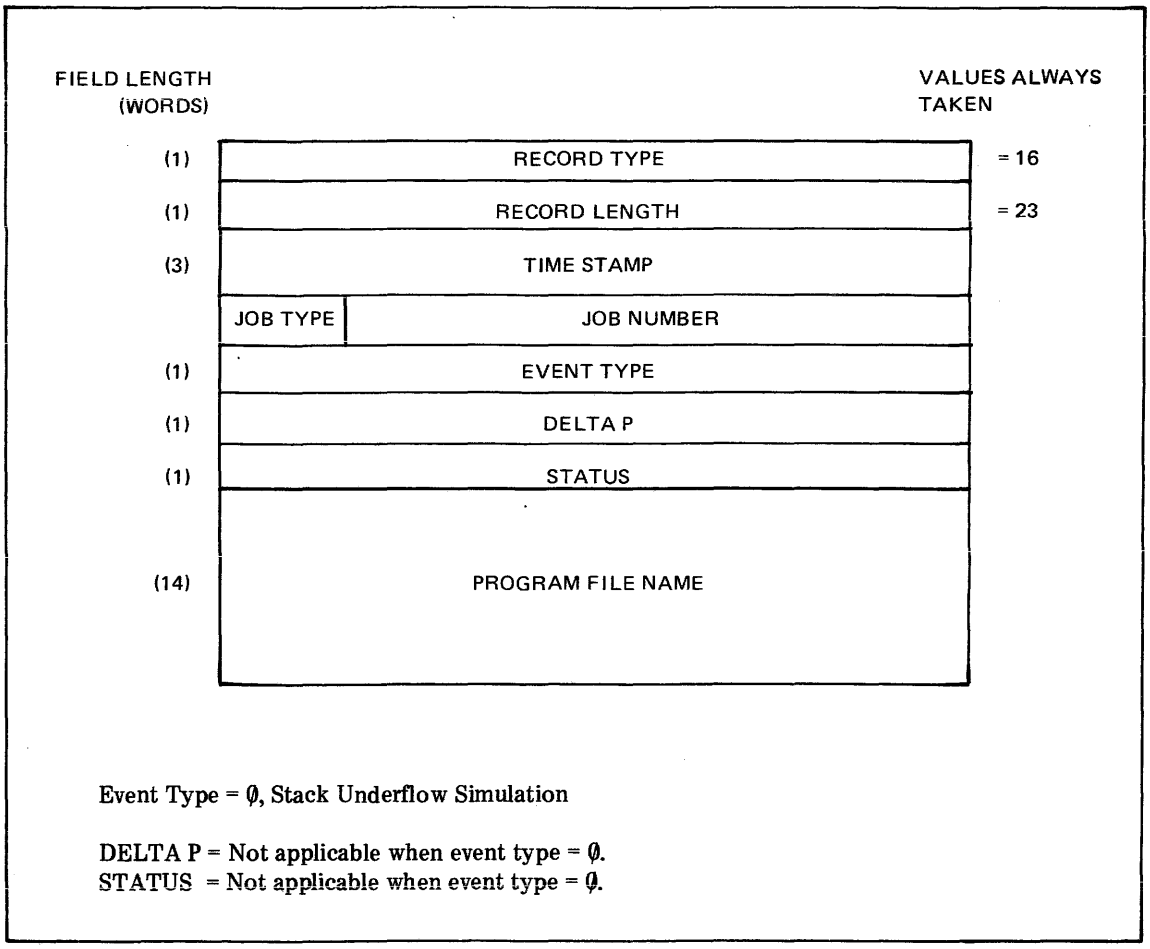


Figure 6-30. Program File Event (Type 16)

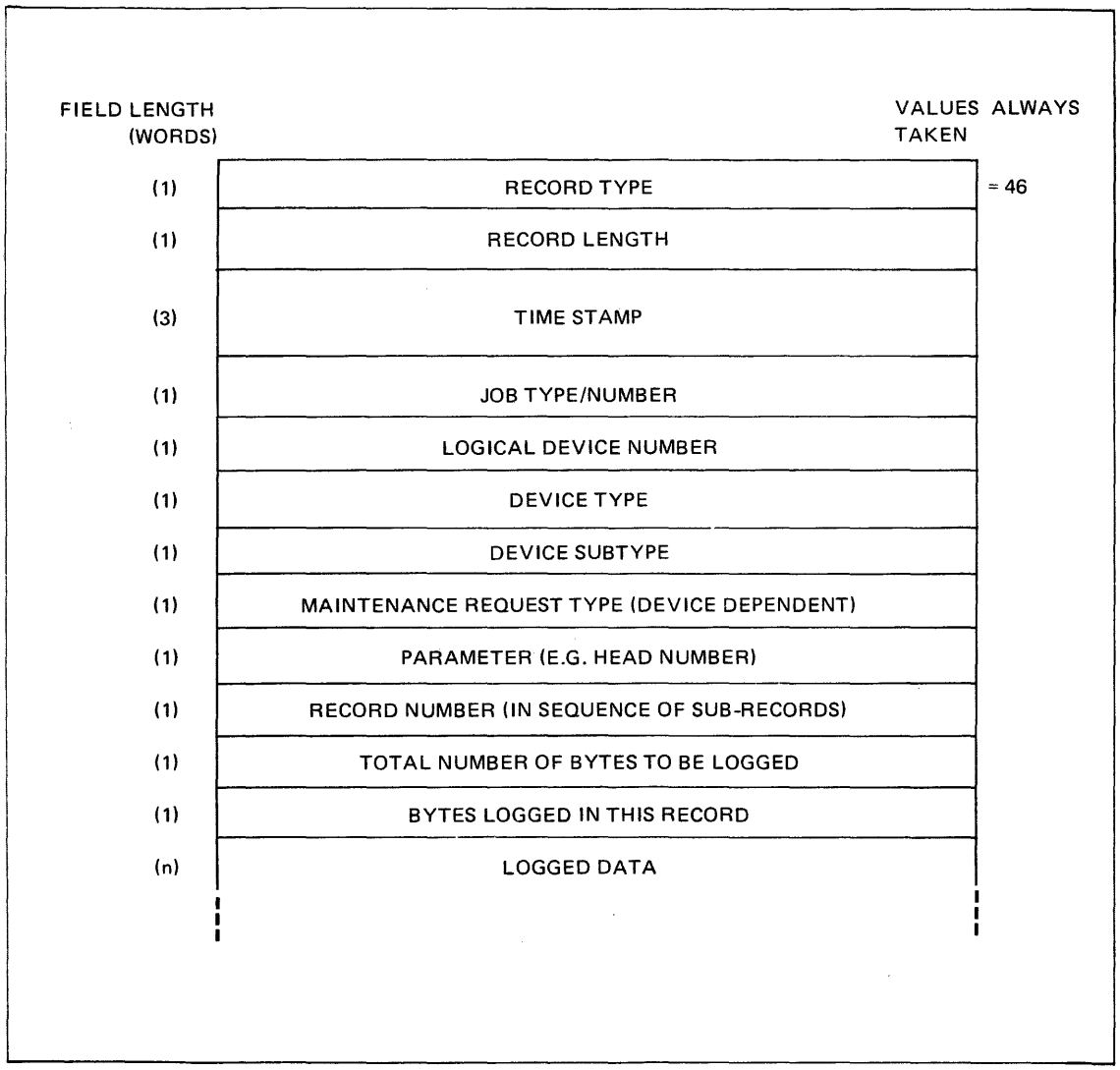


Figure 6-31. MPE Maintenance Request Log Record (Type 46)

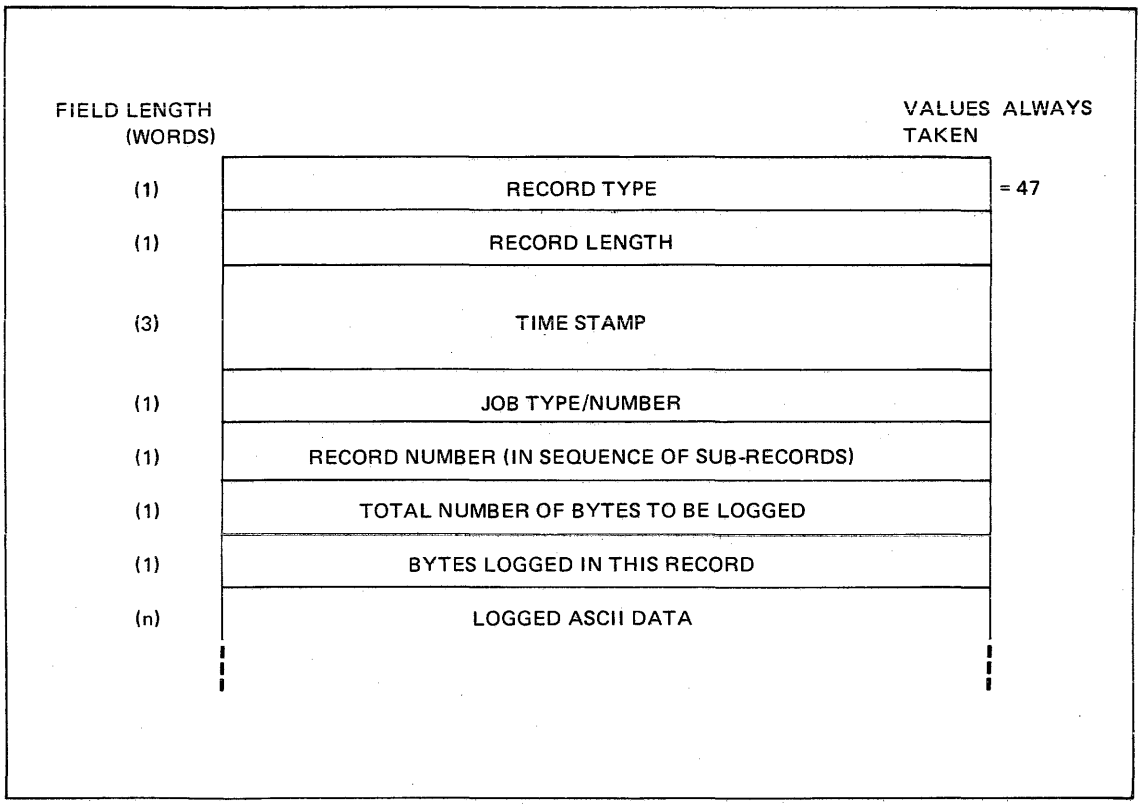


Figure 6-32. Diagnostic Control Unit Log Record (Type 47)

CONSOLE MESSAGES FOR LOG FILES. — Log file errors and status are reported to the system console through messages of the following format:

hh/mm/PIN/message

where

hh	The hour of the day.
mm	The minute of the hour.
PIN	Process Identification Number.
message	The message text.

The message text may consist of any of the following messages:

LOG FILE NUMBER x ON

A new log file (number *x*) has been created. This message always appears before the WELCOME message after coldload. When this message appears while the system is running, it indicates that the previous current log file has been closed. (The actual file name of the log file is *LOGxxxx,* where *xxxx* are four characters representing the same value as *x* in the above message, with leading zeros.)

LOG FILE NUMBER x IS 1/2 FULL

The total space now occupied by the log file data is half the allotted file size.

LOG FILE NUMBER x IS 3/4 FULL

The total space now occupied by the log file data is 3/4 the allotted file size.

LOG FILE NUMBER x ERROR #nn. LOGGING STOPPED

An irrecoverable error occurred, where *nn* is the error number as listed in table 6-8. Logging is disabled until the next system startup.

LOG FILE NUMBER x ERROR #nn. LOGGING SUSPENDED

A recoverable error occurred, where *nn* is the error number as listed in table 6-8. Logging is temporarily suspended pending a :RESUMELOG command.

LOG FILE NUMBER x LOGGING RESUMED

A :RESUMELOG Command was successfully executed.

Table 6-8. Log File Errors

Error No.	Error	Recoverable?
1	Input/output error in accessing the system disc.	No
2	Input/output error in accessing disc log file.	No
21	Data parity error.	No
26	Transmission error.	No
27	Input/output time-out.	No
28	Time-up error or data overrun.	No
29	SIO failure.	No
30	Unit failure.	No
46	Insufficient disc space to create log file.	Yes
47	Input/output error on file lable.	No
57	Virtual memory not sufficient.	No
61	Group (PUB) disc space exceeded in creating log file.	Yes
62	Account (SYS) disc space exceeded in creating log file.	Yes
63	Group disc space exceeded in allocating new extent to the log file.	Yes
64	Account disc space exceeded in allocating new extent to the log file.	Yes
100	A file of the same name as the current log file already exists in the system file directory.	Yes
102	Directory input/output error.	No
103	System directory overflow.	No

MEMORY LOGGING

Memory logging records a history of all memory errors. If error-correcting memory is present on your system, MPE automatically initiates memory error logging during system initialization.

The System Manager uses the MEMLOGAN utility to read the error history records and write them to an output device. (See the *MPE Utilities Manual.)*

USER LOGGING

The MPE user logging facility allows users and subsystems to journalize additions and modifications made to data bases and subsystem files. Specific logging commands and intrinsics are provided to create and maintain user logging files. Logging intrinsics can be used from any subsystem.

If the user is logging to disc, he begins by creating a log file with the BUILD command. For example, to build a log file called DEMOLOG, he enters:

```
:BUILD DEMOLOG;DISC=3000;CODE=LOG
```

No file need be created when the user is logging to a tape file. The GETLOG command is used to initialize and label the new log file:

```
:GETLOG LOGID;PASS=USER;LOG=DEMOLOG,DISC
```

Logging is turned on with the operator LOG command:

```
LOG LOGID,START
```

The intrinsics OPENLOG, WRITELOG, and CLOSELOG can be used programmatically to enter data into this new log file. As with other files on the system, a lockword can be placed on the log file for security. The user need only write a recovery program according to his application to utilize the data stored in the log file.

The maximum number of user logging processes is 64. The maximum number of users for each process is 128.

Additional information about the commands and intrinsics used in user logging is provided in the *MPE Intrinsics Reference Manual* and the *MPE Commands Reference Manual.*

DRIVER NAMES, TYPES, SUB-TYPES AND SIZES

APPENDIX A

Table A-1. Driver Names, Types, Subtypes and Sizes for Series II/III

DEVICE	PART NO.	DRIVER NAME	TYPE	SUBTYPE	RECORD WIDTH (Decimal Words)
Asynchronous Terminal Controller	30032B	IOTERM0	16		40
Hardwired Terminal, speed sensing[1]				1	
Full duplex modem (103, 202T, 212A, or V.21), speed sensing				1	
Asynchronous half-duplex modem (202S or V.23), Data Rate Select ON, speed sensing				2	
Asynchronous half-dupllex modem (202S or V.23), Data Rate Select OFF, speed sensing				3	
Hardwired Terminal or 202T 4-line leased line, speed specified				4	
Full duplex modem (103, 202T, 212A or V.21), speed specified				5	
Asynchronous half-duplex modem (202S or V.23), Data Rate Select ON, speed specified				6	
Asynchronous half-duplex modem (202S or V.23), Data Rate Select OFF, speed specified				7	
Asynchronous Repeater (AR), 30037A (Hardwire environment only) [1]				0,4	
Nine-channel Magnetic Tape Unit					
7970B, 7970E[5]	30215A	IOTAPE0	24	0,8	128
7976A[4,6]		HIOTAPE1	24	1,9	128
Line Printer	30209A	IOLPRT0	32		66/68
2610, 2614				0	
2607				1	
2611, 2613, 2617, 2618, 2619				2	
2617J				3	
2608A				4	
2601		IOTERM0	16	0,1	

Table A-1. Driver Names, Types, Subtypes and Sizes for Series II/III (Continued)

DEVICE	PART NO.	DRIVER NAME	TYPE	SUBTYPE	RECORD WIDTH (Decimal Words)
2631B[2]		IOTERM0	32	14-Hard wired 15-Full duplex modem	66/68
2680A[3]		HIOPPRT0		8	66
Paper Tape Reader (2748B)	30104A	IOPTRD0	9	0	40
Paper Tape Punch (2895A)	30105A	IOPTPN0	34	0	128
Fixed-Head Disc (2660)	30103A	IOFDISC0	1		128
2 megabyte	-001			1	
4 megabyte	-002			2	
Cartridge Disc (7900)	30110A	IOMDISC0	0	2	128
Lower Platter only				1	
Upper Platter only				0	
Disc Drive (7905/7920/7925)	30129A	IOMDISC1	0		128
7905 (Removable Platter)				4	
7905 (Fixed Platter)				5	
7905 (Both Platters)				6	
7905 (System Disc Model)				7	
7920				8	
7925				9	
7933H[4]		HIOMDSC2			
Disc Drive (2888)	30102A	IOMDISC0	0	3	128
Card Reader (2893A)	30106A	IOCDRD0	8	0	40
KATAKANA/Roman Data				1	
Printing Reader/Punch (2894A)	30119A	IOPRPN0	20	0	40

Table A-1. Driver Names, Types, Subtypes and Sizes for Series II/III (Continued)

DEVICE	PART NO.	DRIVER NAME	TYPE	SUBTYPE	RECORD WIDTH (Decimal Words)
Plotter	30126A	IOPLOT0			128
Series 500 Cal Comp			35		
Series 600 Cal Comp			36		
Series 700 Cal Comp			37		
0.010 in.				0	
0.005 in.				1	
0.0025 in.				2	
0.00125 in.				3	
0.002 in.				4	
0.1 mm				5	
0.05 mm				6	
0.025 mm				7	
Programmable Controller (U1)	30361A	IOREM0	23	0	128
DSN/DS	32190A				
Synchronous Single-Line Controller	30055A	CSSBSC0	18		N/A
Switched Line (dialup) with Modem				0	
Nonswitched (leased) Line with Modem; or a hardwired SSLC to INP				1	
Intelligent Network Processor	30010A	IOINP0	17		N/A
Switched (dialup) Line with Modem				0	
Nonswitched (leased) Line with Modem; or a hardwired INP to SSLC				1	
Hardwired INP to INP				3	
Hardwired Serial Interface	30360A	CSHBSC0	19	3	N/A
Communications Driver		IODS0	41		128
No data compression				0	
Data compression				1	
Pseudo Terminal		IODSTRM0	16	0	36

Table A-1. Driver Names, Types, Subtypes and Sizes for Series II/III (Continued)

DEVICE	PART NO.	DRIVER NAME	TYPE	SUBTYPE	RECORD WIDTH (Decimal Words)
DSN/MRJE	32192A	CSSBSC0 and			
Synchronous Single-Line Controller	30055A	CSSMRJE0	18		N/A
Switched (dialup) Line with Modem				0	
Nonswitched (leased) Line with Modem				1	
Pseudo Line Monitor No. 1		IOMRJE0	22	0	40
Pseudo Line Monitor No. 2		IOMRJE1	22	0	40
Pseudo Console		IOMCONS0	22	0	67
Pseudo Line Printer		IOMPNLP0	22	0	67
Pseudo Card Punch		IOMPNLP0	22	0	40
Pseudo Card Reader		IOMRDO0	22	0	40
Intelligent Network Processor	30010A	IOINP0	17		
Switched (dialup) Line with Modem				0	
Nonswitched (leased) Line with Modem				1	
Pseudo Line Monitor No. 1		IOMRJE0	22	0	40
Pseudo Line Monitor No. 2		IOMRJE1	22		40
Pseudo Console		IOMCONS0	22	0	67
Pseudo Line Printer		IOMPNLP0	22	0	67
Pseudo Card Punch		IOMPNLP0	22	0	40
Pseudo Card Reader		IOMDRD0	22	0	40
DSN/RJE	30130E				
Synchronous Single-Line Controller	30055A	CSSBSC0	18		N/A
Switched (dialup) Line with Modem				0	
Nonswitched (private or leased) Line with Modem				1	
Intelligent Network Processor	30010A	IOINP0	17		N/A
Switched (dialup) Line with Modem				0	
Nonswitched (leased) Line with Modem				1	
DSN/MTS	32193A				
Intelligent Network Processor	30010A	IOINP0	17		
Synchronous Switched (dialup) Line with Modem				0	
Synchronous Nonswitched (leased) Line with Modem				1	
Synchronous Hardwired Line				3	
Asychronous Line (hardwired or modem)				7	
OR					
Synchronous Single-Line Controller	30055A	CSSBSC0 and CSSBSC1	18		
Synchronous Switched (dialup) Line with Modem				0	

Table A-1. Driver Names, Types, Subtypes and Sizes for Series II/III (Continued)

DEVICE	PART NO.	DRIVER NAME	TYPE	SUBTYPE	RECORD WIDTH (Decimal Words)
DSN/MTS (continued)					
Synchronous Nonswitched (leased) Line with Modem				1	
Synchronous Hardwired Line				3	
Asychronous Line (hardwired or modem)				7	
Multipoint Supervisor		IOMPS0	16	0	N/A
Multipoint Terminal		IOMPTRM0	16		
Group polling- Display station				0	40
Individual polling- Display station				1	40
Group polling- Data Cap. Terminal				2	N/A
Individual Polling- Data Cap. Terminal				3	N/A
DSN/IMF	32229A				
Intelligent Network Processor	30010A	IOINP0	17		
Synchronous Nonswitched (leased) Line with Modem				1	
DSN/IMF Pseudo Device Driver		IOM3270	22	1	

[1]These terminals should be configured with SUB-TYPE = 1 when hardwired: ASR37, Memorex 1240.

[2]Signed levels of the AR have been tested for compatibility with HP 2640 series terminals.

[3]Configured as remote printer on ATC. See page B-1 for configuration data.

[4]Available on Series III via the HP-IB Interface Module.

[5]For automatic allocation, use subtype 8. (See discussion of automatic allocation in Section 6.)

[6]For automatic allocation, use subtype 9. (See discussion of automatic allocation in Section 6.)

Table A-2. Driver Names, Types, Subtypes and Sizes for Series 30/33/40/44/64

DEVICE	PART NO.	DRIVER NAME	TYPE	SUBTYPE	RECORD WIDTH (Decimal Words)
Asychronous Data Communications Controller Main	30018A	HIOTERM0	16		40
Asychronous Data Communications Controller Extender	30019B	HIOTERM0	16		40
Hardwired Terminal speed sensing[1]				0	
Full duplex modem (103,202T,212A or V.21), speed sensing				1	
Asychronous half-duplex modem (202S or V.23), Data Rate Select ON, speed sensing[2]				2	
Asychronous half-duplex modem (202S or V.23), Data Rate Select OFF, speed sensing[2]				3	
Hardwired Terminal or 202T 4 Line Leased Line, speed specified				4	
2601		IOTERM0		0,1	
2608A		HIOLPRT0	32	4	66/68
2608S		HIOCIPR0	32	9,13[6]	66
2611A/2613A/2617A/2619A		HIOLPRT2	32	2	
2631A		HIOLPRT1	32	5	
2631B[3]		HIOTERM0	32	14-Hard-wired, 15-Full duplex modem	
2680A		HIOPPRT0	32	8	66
Card Reader (2893A)		HIOCDRD0	8	0	40
Nine Channel Magnetic Tape Unit					
7970E		HIOTAPE0	24	0[4],8	128
7976A		HIOTAPE1	24	1[5],9	128
Integrated Cartridge Tape Unit		HIOCTAP0	3	0	
Disc Drives					
7902		HIOFLOP0	2	0	128
9895					

Table A-2. Driver Names, Types, Subtypes and Sizes for Series 30/33/40/44/64 (Continued)

DEVICE	PART NO.	DRIVER NAME	TYPE	SUBTYPE	RECORD WIDTH (Decimal Words)
7906		HIOMDSC1		10	
7906 (removable platter)					
7906 (fixed platter)				11	
7906 (both platters)				12	
7911		HIOMDSC2	3	1	
7912		HIOMDSC2	3	2	
7920		HIOMDSC1	0	8	
7925		HIOMDSC1	0	9	
7933		HIOMDSC2	3	8	
DSN/RJE	30130E				
Intelligent Network Processor	32002A	IOINP0	17		N/A
Switched (dialup) Line with Modem				0	
Nonswitched (leased) Line with Modem				1	
DSN/DS	32190A				
Intelligent Network Processor	30020A	IOINP0	17		N/A
Switched (dialup) Line with Modem				0	
Nonswitched (leased) Line with Modem; or Hardwired INP to SSLC				1	
Hardwired INP to INP				3	

[1]These terminals should be configured with SUBTYPE = 1 when hardwired: ASR37, Memorex 1240

[2]Not supported on Series 44 via the DSN/ADCC.

[3]Configured as remote printer on DSN/ADCC. See page B-1 for configuration data.

[4]Available via the HP-IB Interface Module.

[5]For automatic allocation, use subtype 8. See discussion on automatic allocation in Section 6.

[6]Subtype 9 is for feature access, 13 is for transparent access.

DRIVER CONFIGURATION DATA

APPENDIX B

For the following Hewlett-Packard-supported devices and drivers, input the information described below into steps 9 through 48 and step 138 of the Configurator/user dialogue. Driver names are given in parenthesis, first for Series II/III, then for Series 30/33/40/44/64.

Table B-1. Asynchronous Terminal Controller and Asynchronous Data Communications Controller Driver (IOTERMØ, HIOTERMØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
9	DRT #?	For Series 30/33/40/44/64 system console must be DRT 8; see 6-22
10	UNIT #?	For Series 30/33/40/44/64, all terminals must be UNIT 0; see 6-22.
12	TYPE?	16
13	SUBTYPE?	(See Appendix A)
14	TERM TYPE?	(Varies with terminal — See below)
15	SPEED IN CHARACTERS PER SECOND?	10, 14, 15, 30, 60, 120, or 240 (for Series 30/33/40/44/64 systems, 480 and 960 are also available; 14 and 15 are not available)
34	REC WIDTH?	(Varies with terminal)
35	OUTPUT DEVICE?	(ldn or class name of listing device)
36	ACCEPT JOBS/SESSIONS?	YES
37	ACCEPT DATA?	YES
38	INTERACTIVE?	YES
39	DUPLICATIVE?	YES
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	(IOTERM0, HIOTERM0)
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	(IOTERM0, HIOTERM0), <prepared file name>

TERMINAL TYPES

0 HP 2749B, ASR33
1 ASR37
2 ASR35
3 Execuport 300
4 HP 2600A, Datapoint 3300
5 Memorex 1240
6 HP 2762A/B, GE Terminet 300 and 1200
9 HP 2615A Beehive Mini Bee
10 HP 2621A/P, 3075A, 3076A, 3077A, 2640A/B, 2641A, 2644A, 2645A, 2647A, 2648A, Character Mode or program controlled
*11 HP 2640A/B, 2641A, 2644A, 2645A, 2647A, Character mode and block page mode without program control
12 HP 2645K Katakana Roman Data Terminal
13 Terminal connected to a switching network or another computer
15 HP 2635A Printing Terminal (8-bit protocol for second character set)
16 HP 2635A Printing Terminal (7-bit protocol for standard character set)
19 HP 2631B Remote Printer

NOTE: On Seires 30/33 only terminal types 4, 6, 9, 10, 11, 12, 15, 16 and 19 are available.

* On Series 40/44, term type 4, 6, 9, 10, 12, 15, 16 and 19 are available, not term type 11.

Table B-1A. Advanced Terminal Processor Driver (HIOTERM1)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
9	DRT #?	The last digit of a DRT number must be 0, in octal.
10	UNIT #?	The unit number depends on the hardware configuration of the terminal. It is a number between 0 and 95.
12	TYPE?	16
13	SUBTYPE?	(See Appendix A)
14	TERM TYPE?	Varies with terminal. ATP will not support term types 0, 1, 2, 3, 4, 7, 8, 11, 14 or 17.
15	SPEED IN CHARACTERS PER SECOND?	10, 30, 60, 120, 240, 480 or 960.
34	REC WIDTH?	(Varies with terminal.)
35	OUTPUT DEVICE?	(ldn or class name of listing device.)
36	ACCEPT JOBS/SESSIONS?	YES
37	ACCEPT DATA?	YES
38	INTERACTIVE?	YES
39	DUPLICATIVE?	YES
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	HIOTERM1
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOTERM1, <prepared file name>

NOTE: Advanced Terminal Processor (ATP) is available only with Series 44, with direct connect, and Series 64, with direct connect and with modems.

Table B-2. 7970B, E Nine-Channel Magnetic Tape Unit Driver (IOTAPE0, HIOTAPE0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	24
13	SUBTYPE?	0 (8 for automatic allocation; see section 6.)
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	YES or NO
37	ACCEPT DATA?	YES or NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES or NO
42	INPUT OR OUTPUT?	IN (asked only if initially spooled)
43	DRIVER NAME?	(IOTAPE0, HIOTAPE0)
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	(IOTAPE0, HIOTAPE0) <prepared file name>

Table B-2A. 7976A Nine-Channel Magnetic Tape Driver (HIOTAPE1)*

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	24
13	SUB TYPE?	1 (9 for automatic allocation; see section 6.)
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOB/SESSIONS?	YES or NO
37	ACCEPT DATA?	YES or NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES or NO
42	INPUT or OUTPUT?	IN (asked only if initially spooled)
43	DRIVE NAME?	HIOTAPE1
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOTAPE1, < prepared file name >

* Available on Series III via the HPIB Interface Module. Not available on Series II.

Table B-2B. Integrated Cartridge Tape Unit Driver (HIOCTAP0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	3
13	SUBTYPE?	0
34	RECORD WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	HIOCTAP0
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOCTAP0 <prepared file name>

Table B-3. Fixed-Head Disc Driver (IOFDISC0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	1
13	SUB TYPE?	(See Appendix A)
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOFDISCØ
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOFDISCØ, <prepared file name>

Table B-4. 7900A Cartridge Disc Driver (IOMDISC0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	0
13	SUB TYPE?	(See Appendix A)
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOMDISCØ
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOMDISCØ, <prepared file name>

Table B-5. 7902 Flexible Disc Driver (HIOFLOPØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	2
13	SUB TYPE?	0
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	HIOFLOPØ
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOFLOPØ, <prepared file name>

NOTE: The default DRT for the system disc for Series 30/33/40/44/64 is 16. Default for the system disc for Series II/III systems is 4.

Table B-6. 7905A, 7906A* , 7920A, or 7925 Disc Driver (IOMDISC1, HIOMDSC1)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	0
13	SUBTYPE?	(See Appendix A)
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	YES or NO
43	DRIVER NAME?	(IOMDISC1, HIOMDSC1)
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	(IOMDISC1, HIOMDSC1) < prepared file name >

* Not available on Series II/III.

Table B-6A. 7911, 7912, or 7933 Disc Driver (HIOMDSC2)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	3
13	SUBTYPE?	1 for 7911, 2 for 7912, 8 for 7933
34	RECORD WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	HIOMDSC2
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOMDSC2 <prepared file name>

Table B-7. 9895 Flexible Disc Drive (HIOFLOPØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	2
13	SUBTYPE?	0
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	YES or NO
43	DRIVER NAME?	HIOFLOPØ
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOFLOPØ, < prepared file name >

Table B-8. Disc File Driver (IOMDISC0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	0
13	SUB TYPE?	3
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOMDISC0
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOMDISC0, <prepared file name>

Table B-9. Card Reader Driver (IOCDRD0, HIOCDRD0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	8
13	SUB TYPE?	0 (1 for KATAKANA translation)
34	REC WIDTH?	40
35	OUTPUT DEVICE?	(ldn or classname of listing device)
36	ACCEPT JOBS/SESSIONS?	YES
37	ACCEPT DATA?	YES
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES or NO
43	DRIVER NAME?	(IOCDRD0, HIOCDRD0)
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	(IOCDRD0, HIOCDRD0) <prepared file name>

* The HP 2893A Card Reader is available only on Series 30/33/40/44/64. It is supported on a dedicated GIC only.

Table B-10. Line Printer Driver (IOLPRTØ, HIOLPRTØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	32
13	SUB TYPE?	(See Appendix A)
34	REC WIDTH?	66
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES or NO
43	DRIVER NAME?	(IOLPRTØ, HIOLPRTØ), see Table A-2, A-5
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOLPRTØ, HIOLPRTØ, <prepared file name>

Table B-11. 2631A Line Printer Driver (HIOLPRT1)*

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	32
13	SUBTYPE?	(See Appendix A)
34	REC WIDTH?	66
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES or NO
43	DRIVER NAME?	(IOLPRT1, HIOLPRT1)
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	(IOLPRT1, HIOLPRT1) <prepared file name>

* Available only on Series 30/33/40/44/64 Systems.

Table B-12A. 2608S Line Printer Driver (HIOCIPRØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	32
13	SUBTYPE?	9 or 13 (see discussion on transparent feature access)
34	RECWIDTH?	66
35	OUTPUT DEVICE?	Ø
36	ACCEPT JOB/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES or NO
43	DRIVER NAME?	HIOCIPRØ
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOCIPRØ , < prepared file name>

Table B-11A. 2631B Line Printer Driver* (IOTERMØ, HIOTERMØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
9	DRT#?	DRT #
10	UNIT #?	ATC portion unit Series II,III; 0 on Series 30/33/40/44/64
12	TYPE?	32
13	SUBTYPE?	14 - Hardwired 15 - Full duplex modem
14	TERM TYPE?	19
15	SPEED IN CHARACTERS PER SECOND?	Any valid speed, 120
34	REC WIDTH?	66
35	OUTPUT DEVICE?	Ø
36	ACCEPT JOB/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES/NO
43	NAME DRIVER?	(IOTERMØ, HIOTERMØ)
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	(IOTERMØ, HIOTERMØ), < prepared file name >

* For each spooled 2631B, an additional 16 IOQ's should be configured in the :SYSDUMP System Tables changes section.

Table B-12. 2613/2617/2619 Line Printer Driver (HIOLPRT2)*

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	32
13	SUBTYPE?	2
34	REC WIDTH?	66
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED	YES or NO
43	DRIVER NAME?	HIOLPRT2
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOLPRT2 <prepared file name >

* For use on Series 30/33/40/44/64, see A-5.
Series II/III Line Printer Drivers, see IOLPRT0, A-1.

Table B-12B. 2680A Laser Printing System Driver (HIOPPRTØ)*

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	32
13	SUBTYPE?	8
34	REC WIDTH?	66
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES
42	DRIVER NAME	HIOPPRTØ
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	HIOPPTRØ, < prepared file name >

* Available on Series III via the HPIB Interface Module.

Table B-13. Paper Tape Reader Driver (IOPTRDØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	9
13	SUB TYPE?	0
34	REC WIDTH?	40
35	OUTPUT DEVICE?	(ldn or classname of listing device)
36	ACCEPT JOBS/SESSIONS?	YES
37	ACCEPT DATA?	YES
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOPTRD0
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOPTRD0, <prepared file name>

Table B-14. Paper Tape Punch Driver (IOPTPNØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	34
13	SUB TYPE?	0
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOPTPN0
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOPTPN0, <prepared program file name>

Table B-15. Plotter Driver (IOPLOTØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	35
13	SUB TYPE?	(0 through 7 — see Appendix A)
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES or NO
43	DRIVER NAME?	IOPLOT0
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOPLOT0, <prepared program file name>

Table B-16. Printing Reader/Punch Driver (IOPRPNØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	20
13	SUB TYPE?	0
34	REC WIDTH?	40
35	OUTPUT DEVICE?	(ldn or classname of listing device)
36	ACCEPT JOBS/SESSIONS?	YES
37	ACCEPT DATA?	YES
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES or NO
42	INPUT OR OUTPUT?	IN or OUT
43	DRIVER NAME?	IOPRPN0
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOPRPN0, <prepared program file name>

Table B-17. Programmable Controller (UI) Driver (IOREMØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
12	TYPE?	23
13	SUB TYPE?	0
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOREM0
138	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOREM0, <prepared program file name>

Table B-18. Synchronous Single-Line Controller (CSSBSCØ, CSSBSC1, CSSMRJEØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE	
10	UNIT#?	0	
11	SOFTWARE CHANNEL#?	0	
12	TYPE?	18	
13	SUB TYPE?	0 or 1	(DSN/DS, DSN/RJE, DSN/MRJE)
		0, 1, 3, or 7	(DSN/MTS)
17	PROTOCOL?	1	(DSN/DS, DSN/RJE, DSN/MTS)
		1 - 255	(DSN/MRJE - any value, not used)
18	LOCAL MODE?	1 or 2	(DSN/DS, DSN/RJE)
		1	(DSN/MTS)
		1 - 15	(DSN/MRJE - any value, not used)
19	TRANSMISSION CODE?	1, 2, or 3	(DSN/DS, DSN/RJE)
		1 - 63	(DSN/MRJE, DSN/MTS - any value, not used)
20	RECEIVE TIMEOUT?	0 - 32000	(DSN/DS, DSN/RJE, DSN/MTS)
		Return	(DSN/MRJE) Response not used. Default 20 seconds.
21	LOCAL TIMEOUT?	0 - 32000	(DSN/DS, DSN/RJE, DSN/MTS)
		Return	(DSN/MRJE) Response not used. Default 60 seconds.
22	CONNECT TIMEOUT?	0- 32000	(DSN/MTS, DSN/RJE DSN/DS - 300 seconds, recommended)
		Return	(DSN/MRJE - Response not used. Default 900 seconds.
23	DIAL FACILITY?	YES or NO	(DSN/DS, DSN/RJE, DSN/MTS)
		Return	(DSN/MRJE)
24	ANSWER FACILITY?	YES or NO	(DSN/DS, DSN/RJE, DSN/MTS)
		Return	(DSN/MRJE)
25	AUTOMATIC ANSWER?	YES or NO	(DSN/DS, DSN/RJE, DSN/MTS)
		Return	(DSN/MRJE)
26	DUAL SPEED?	YES	(European modems)
		NO	(All others and DSN/MTS)
27	HALF SPEED?	YES or NO	
28	SPEED CHANGEABLE?	YES or NO	
29	TRANSMISSION SPEED?	250, 300, 600, or 1200	
30	TRANSMISSION MODE?	0 (Full duplex), 1 (Half duplex)	

Table B-18. Synchronous Single-Line Controller (CSSBSCØ)
(Continued)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE	
31	PREFERRED BUFFER SIZE?	0 - 4095	(DSN/DS - 1075 recommended)
		n > 0	(DSN/MRJE, DSN/RJE Any value, response not used
		250-2000	(DSN/MTS) - 500 recommend
32	DRIVER CHANGEABLE?	NO	(DSN/DS, DSN/RJE)
		YES	(DSN/MTS, DSN/MRJE)
33	DRIVER OPTIONS?	0	
43	DRIVER NAME?	CSSBSC0	
44	PHONELIST?	YES or NO	(DSN/DS, DSN/RJE, DSN/MTS)
		NO	(DSN/MRJE)
45	PHONE NUMBER?	number	(A telephone number - not more than 20 characters)
46	LOCAL ID SEQUENCE?	Return	(DSN/MRJE, DSN/MTS, DSN/RJE)
		Return or ID	(DSN/DS)
47	REMOTE ID SEQUENCE?	Return	(DSN/MRJE, DSN/MTS, DSN/RJE)
		Return or ID	(DSN/DS)
48	DEVICE CLASSES?	Class Name or Return	(DSN/DS, DSN/MRJE, DSN/MTS)
		RJLINE*****	(DSN/RJE - Additional classes are optional)
67	ADDITIONAL DRIVER CHANGES?	NO	(DSN/DS, DSN/RJE)
		YES	(DSN/MTS, DSN/MRJE)
71	ADD DRIVERS?	YES	(DSN/MRJE, DSN/MTS)
72	DRIVER NAME?	CSSMRJEØ	(DSN/MRJE)
		CSSBSC1	(DSN/MTS)

NOTES

* Question asked only if subtype is 0.
** Question asked only if response to DUAL SPEED? is YES.
*** Questions asked only if response to DUAL SPEED? is NO.
**** For hardwired SSLC to INP connections, use subtype 1 and transmission mode 1.
***** DSN/RJE defaults to RJLINE as the device class if NO device is specified in the :RJLINE. command.

Table B-19. DSN/DS Communications Driver (IODS0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
10	UNIT #?	0
11	CHANNEL #?	0
12	TYPE?	41
13	SUBTYPE?	0 No data compression 1 Data compression
34	REC WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IODS0

Table B-20. DSN/DS Pseudo Terminal (IODSTRM0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
10	UNIT#?	0
11	CHANNEL#?	0
12	TYPE?	16
13	SUBTYPE?	0 (8 for 3000-1000 V/PLUS)
14	TERMTYPE?	carriage return
15	SPEED IN CHARACTERS PER SECOND?	carriage return
34	REC WIDTH?	40
35	OUTPUT DEVICE?	same as the input device; that is, ldev of Pseudo Terminal
36	ACCEPT JOBS/SESSIONS?	YES
37	ACCEPT DATA?	NO
38	INTERACTIVE?	YES
39	DUPLICATIVE?	YES
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IODSTRM0

Table B-21. Hardwired Serial Interface (CSHBSCØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
10	UNIT#?	0
11	CHANNEL#?	0
12	TYPE?	19
13	SUBTYPE?	3
16	PORTMASK?	8 (chan0), 4 (chan1), 2 (chan2), or 1 (chan3)
17	PROTOCOL?	1
18	LOCAL MODE?	1
19	TRANSMISSION CODE?	2
20	RECEIVE TIMEOUT?	0-32000 or return
21	LOCAL TIMEOUT?	0-32000 or return
22	CONNECT TIMEOUT?	0-32000 or return (300 recommended
28	SPEED CHANGEABLE?	YES
29	TRANSMISSION SPEED?	250000 or 125000
30	TRANSMISSION MODE?	0
31	PREFERRED BUFFER SIZE?	0-4095 (4095 recommended)
32	DRIVER CHANGEABLE?	NO
33	DRIVER OPTIONS?	0
43	DRIVER NAME?	CSHBSCØ

Table B-22. DSN/MRJE Pseudo Devices (IOMRJE0, IOMRJE1, IOMCONS0, IOMPNLPØ, IOMRDRØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSES
10	UNIT#?	0 (Line monitor no. 1) 1 (Line monitor no. 2) 2 (console) 3 thru 9 (line printers) 10 thru 16 (card punches) 17 thru 23 (card readers)
11	CHANNEL#?	0
12	TYPE?	22
13	SUBTYPE?	0
34	RECORD WIDTH?	40 (line monitors, card readers and card punches) 67 (line printers, console)
35	OUTPUT DEVICE?	0
36	ACCEPT JOB/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	YES (first pseudo card reader only) NO (all other pseudo devices)
42	INPUT OR OUTPUT?	OUTPUT (one pseudo card reader only)

Table B-22. DSN/MRJE Pseudo Devices (IOMRJE0, IOMRJE1, IOMCONS0, IOMPNLPØ, IOMDRDØ) (Continued)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
43	DRIVER NAME?	IOMRJE0 (line monitor no. 1) IOMRJE1 (line monitor no. 2) IOMCONS0 (console) IOMPNLP0 (line printers and punches) IOMRDR0 (card readers)
48	DEVICE CLASS?	classname or return (spooled card reader) Note: DSN/MRJE expects a classname of MDRD1. return (all other pseudo devices)

Table B-23. Multipoint Terminal (IOMPTRMØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
10	UNIT#?	nnn (Unique group/device ID)
11	CHANNEL#?	0
12	TYPE?	16
13	SUBTYPE?	0 Subtype? 0 Group polling, display stations. 1 Individual polling, display stations. 2 Group polling, data cap terminals. 3 Individual polling, data cap terminals.
14	TERM TYPE?	14
15	SPEED IN CHARACTERS PER SECOND?	CR
34	RECORD WIDTH?	40*
35	OUTPUT DEVICE?	(ldn or classname)
36	ACCEPT JOBS/SESSIONS?	YES*
37	ACCEPT DATA?	YES*
38	INTERACTIVE?	YES
39	DUPLICATIVE?	YES
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOMPTRMØ
48	DEVICE CLASSES?	classname or return

* For 307X terminal, response is not used.)

Table B-24. Multipoint Supervisor (IOMPSØ)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
10	UNIT#?	0
11	CHANNEL#?	0
12	TYPE?	16
13	SUBTYPE?	0
14	TERM TYPE?	14
15	SPEED IN CHARACTERS PER SECOND?	CR
34	RECORD WIDTH?	40 (any non-zero value acceptable)
35	OUTPUT DEVICE?	(ldn or classname)
36	ACCEPT JOBS/SESSIONS?	CR
37	ACCEPT DATA?	CR
38	INTERACTIVE?	CR
39	DUPLICATIVE?	CR
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOMPSØ
48	DEVICE CLASSES?	classname or return

Table B-25. Intelligent Network Processing (IOINP0)

Note: The DS/INP supports DSN/DS, DSN/RJE, DSN/MRJE, DSN/MTS, and DSN/IMF

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
10	UNIT #?	0
11	SOFTWARE CHANNEL #?	0
12	TYPE?	17
13	SUB TYPE?	0 or 1 (DSN/RJE) 0, 1, or 3 (DSN/DS) 1 for DSN/IMF
20	RECEIVE TIMEOUT?	0-32000 or return (default - 20 seconds)
21	LOCAL TIMEOUT?	0-32000 or return (default - 60 seconds)
22	CONNECT TIMEOUT?	0-32000 or return (300 recommended for DS and RJE) (default - 900 seconds recommended for IMF)
23	DIAL FACILITY?	YES or NO (Asked for subtype 0 only)
24	ANSWER FACILITY?	YES or NO (Asked for subtype 0 only)
25	AUTOMATIC ANSWER?	YES or NO (Asked for subtype 0 only)
26	DUAL SPEED?	YES or NO
27	HALF SPEED?	YES or NO (Asked if DUAL SPEED? was YES only)
28	SPEED CHANGEABLE?	YES or NO (Asked only if DUAL SPEED? was NO)
29	TRANSMISSION SPEED?	250, 300, 600, 1200, 2400, 3600, 4800, 7000 characters per second
30	TRANSMISSION MODE?	(0 = Full Duplex, 1 = Half Duplex)
31	PREFERRED BUFFER SIZE?	0-1024 (DS/3000; 1024 recommended) Note: RJE overrides this option)
32	DRIVER CHANGEABLE?	No
33	DRIVER OPTIONS	0
43	DRIVER NAME	IOINPØ
44	PHONE LIST	YES or NO (Asked for subtype 0 only)
46	LOCAL ID SEQUENCE?	ID or return
47	REMOTE ID SEQUENCE	ID or return (Asked for subtype 0 only) (DSN/RJE, response not used)
48	DEVICE CLASSES?	Classname list or return. DSN/RJE: RJELINE (additional names optional)

*For hardwired INP to SSLC connections, use subtype 1 and transmission mode 1.

For hardwired INP to INP connections, use subtype 3 and transmission mode 0.

**DSN/RJE default to RJLINE as the device class if NO device is specified in the : RJLINE command.

Table B-26. DSN/IMF Pseudo Device (IOM3270)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSES
10	UNIT #?	0
11	SOFTWARE CHANNEL #?	0 (Response not used by IML)
12	TYPE?	22
13	SUBTYPE?	1
34	RECORD WIDTH?	128
35	OUTPUT DEVICE?	0
36	ACCEPT JOBS/SESSIONS?	NO
37	ACCEPT DATA?	NO
38	INTERACTIVE?	NO
39	DUPLICATIVE?	NO
41	INITIALLY SPOOLED?	NO
43	DRIVER NAME?	IOM3270
48	DEVICE CLASSES?	CLASSNAME OR RETURN

RECOMMENDED PARAMETER SETTINGS

APPENDIX C

CONFIGURATION GUIDELINES

Choosing an appropriate set of configuration parameters is an important part of tuning an HP 3000. In general, the various parameters should be as small as possible while still providing sufficient resources to process the required work load. The following guidelines are offered to help the system manager make an intelligent choice of parameters for his or her particular system.

The quidelines presented here assume that users are using only the standard capabilities; sites using the process handling or extra data segment capabilities should make additional allowances for the extra resources required. The table at the end of the parameter descriptions gives suggested values for a 16 user system. It also indicates the memory requirements for each table, and whether it is permanently resident. The following terms are used:

Jobs — The maximum number of jobs that will be run at any given time.

Sessions — The maximum number of sessions that will be run at any given time.

Users — The maximum number of jobs and sessions that will be run at any given time.

Note that Users, as defined here, may be less than the sum of Jobs plus Sessions. For example, during the day the system manager allows 1 job and 16 sessions, in the evening 3 jobs and 10 sessions, and at night 6 jobs and 5 sessions.
Jobs = the maximum jobs = 6
Sessions = the maximum sessions = 16
Users = the maximum at any one time = 17

Terminals — The number of configured terminals.

1. MAX # OF OPEN SPOOLFILES
   This is the maximum number of spool files that may be in the OPENED state at any one time; it does not affect spool files in the ACTIVE or READY states. When a spool file is opened, MPE creates a "virtual device" of the required type by filling in an unused logical device entry with the appropriate values. In essence, this parameter tells MPE how many logical device number to reserve for spooling.

   A spooled or streamed job requires at least two opened spoolfiles, one for $STDIN and one for $STDLIST. In addition, each user requires an open spoolfile for each access to a spooled device.

2. CST
   The Code Segment Table (CST) is used for all permanently allocated MPE and subsystem code segments (those listed in the LOADMAP) plus all code segments that come from any segmented library (SL). MPE needs a minimum of 93 entries to operate. Facilities doing program development, and those which have large SLs, need more CST entries. If the CST is underconfigured, some programs will not run when the system is heavily loaded. If overconfigured, main memory space is wasted. It is recommended that 192 entries, the maximum number allowed, be used in most cases.

3. EXTENDED CST

This table is used for all program code segments which are assigned dynamically via the :RUN command or allocated permanently via the :ALLOCATE command. MPE needs at least 32 entries to operate. Large programs with many code segments require more table entries. The number of additional XCST entries is dependent on the number of users and the number and sizes of different applications which are running. If the extended CST is underconfigured, some programs will not run when the system is heavily loaded. If over-configured, main memory is wasted. (The :ALLOCATE command should be used cautiously if there is a limited amount of space in the XCST).

4. DST

The Data Segment Table is used to reference all data segments, including MPE and user data stacks, etc. The recommended number of DST entries is (8 x # of users) + 64. Programs which use many extra data segments or which have many open files require more entries. If underconfigured, some programs may not run when the system is heavily loaded. If overconfigured, main memory is wasted.

5. PCB

The Process Control Block contains the status of each process on the system. Each entry consumes 16 words and there is 1 entry per process. The recommended configuration is: 14 entries + # spooled devices + # of DS lines + # of MRJE lines + # user logging processes + (2 x # users). If process handling is used, add more entries to the above configuration value. If underconfigured, system failure may result. If overconfigured, main memory is wasted.

6. DISC REQUEST TABLE

The Disc Request Table (DRQ) is used to hold request parameters and queries for pending I/O's. Allow 60 entries for the average system with 16 users and 4 discs. In general, the configuration value will be a function of the number of processes and the number of discs. If underconfigured, disc requests will wait for free entries, and performance will be affected.

7. I/O QUEUE

This single queue is used to hold all I/O requests on the system except disc requests. Allow 4 entries for each system user. If underconfigured, non-disc requests will wait for free entries, and performance will be affected.

8. TERMINAL BUFFERS PER PORT

Terminal buffers are used for all buffered terminal I/O. On Series II/III systems, configure 6 buffers per active terminal user. On the Series 30/33/40/44/64 system, use 4 buffers per active terminal user. If mostly block mode applications are to be run on the system (i.e. V/3000), configure the maximum 256 buffer on all 3000's. If underconfigured, terminal response time will be slow and the cursor movement will be erratic when writing data to the screen. If overconfigured, main memory space is wasted.

Note: Terminal buffers are shared among all terminals. Specifying a value for terminal buffers "per port" will not necessarily limit a given terminal to that number of buffers.

9. SYSTEM BUFFERS

System buffers are used by the message system to handle console messages and by the communications subsystems. The average configuration size is 8 buffers. If the system has many interactive users and a large number of streamed jobs, you will need a larger number of buffers to handle the logons and logoffs. Heavy usage of :TELLOP or :TELL, or using private volumes without issuing the console command :VMOUNT on, auto will require more system buffers. One additional buffer is needed for each multipoint line. If underconfigured, system response may be very slow. If overconfigured, main memory space is wasted.

10. SWAP TABLE

The suggested number of entries is 5 times the number of PCB entries configured in the system, or 6 times the number of PCB entries if a large number of extra data segments is used. If underconfigured, System Failure (602) may occur. If overconfigured main memory space is wasted. The Swap Table is used by the Memory Manager to determine which segments need to be brought into memory in order for an absent process to run.

11. PRIMARY MESSAGE TABLE

The Primary Message Table is used to buffer message contents for pending messages transmitted within the system using the internal message facility during the execution of critical areas of system code. The suggested number of entries is 25.

Note: The number of primary message table entries configured should be greater than the configured number of system buffers.

12. SECONDARY MESSAGE TABLE

The Secondary Message Table is used to buffer message contents for pending messages transmitted within the system using the internal message facility. If the Secondary Message Table becomes full, processes requiring it will be impeded until entries become available. The suggested number of entries is 25.

13. SPECIAL REQUEST TABLE

The Special Request Table is used to temporarily buffer the parameters for segment expansion and to form the queue for devices waiting for a segment to arrive in memory. The suggested number of entries is 25.

14. ICS

The Interrupt Control Stack is used to process all interrupt requests. Allow at least 768 words. If underconfigured, the system may fail.

15. UCOP REQUEST QUEUE

This is the request queue for the User Controller Process. UCOP is responsible for deleting processes, changing priorities, and expanding or contracting stacks. Allow approximately 2 * Users — add more if process handling is used.

16. TIMER REQUEST LIST

The Timer Request List is used for the memory logging facility, calls to PAUSE, timed terminal I/O, and various other timings such as modem turn around. Allow approximately 1 * Terminals + 1 * Users.

17. BREAKPOINT TABLE

The Breakpoint Table is used to hold the information needed for DEBUG breakpoints. Allow at least 16 entries to aid in troubleshooting, and add more if DEBUG is used extensively.

18. MAX # OF USER LOGGING PROCESSES

Normally one process per application is sufficient. The maximum number of user logging processes is 64.

19. MAX # OF USERS PER LOGGING PROCESS

The number of users who need to access logging file. The maximum number of users for each process is 128.

20. # OF RINS

This parameter is the total number of Resource Identification numbers available to the system. It includes global rins, local rins, and file rins which are used whenever dynamic file locking is requested. The number of rins needed depends on the users' applications, but there should be at least one per user to allow for the dynamic file locking used by subsystems such as the SEGMENTER and SYSDUMP.

21. # OF GLOBAL RINS

This parameter is used to allocate the space needed to save the user names, account names, and passwords associated with global rins. Since this parameter can only be changed on a reload, it is a good idea to allow a few more than are expected to be needed.

22. # OF SECONDS TO LOGON

Whenever a terminal is speed sensed to begin a logon, MPE starts a logon timeout. If the time expires before the logon is complete, the port is reset to its initial state. This is particularly important with modems to prevent a wrong number from hanging a port indefinitely. The time should be as short as possible so that improper logons can be reset promptly, but it must be long enough so that the slowest typist can logon successfully. Sixty seconds is usually adequate.

23. MAX # OF CONCURRENTLY RUNNING SESSIONS
24. MAX # OF CONCURRENTLY RUNNING JOBS

These parameters limit the number of interactive sessions and batch jobs allowed on the system at one time. The number of sessions is limited by the number of configured terminals and by the number of PCB entries. There is at least 1 PCB entry for each session or job. Setting the job and session limits is important in load management, and will vary according to the applications. Generally, the more user/jobs, the greater the system load. Do not underconfigure the session and job values, control them by using the :LIMIT command up to the value specified in the configuration. The :LIMIT command may be used to dynamically adjust job/session mix up to the configured maximums.

25. DEFAULT JOB CPU TIME LIMIT

This parameter may be used to establish a default limit on the number of CPU seconds that a job may use. This default will be used whenever the TIME=cpusecs parameter is omitted from a JOB statement. If this parameter is 0, no default limit will be imposed.

26. LOG FILE RECORD SIZE (SECTORS)
27. LOG FILE SIZE (RECORDS)

These parameters specify how log files are to be built. The values given in the table are suitable for all systems.

28. VIRTUAL MEMORY

Virtual memory may be allocated on any system domain disc volume. This area is used only for data segments since code segments are read directly from the program file or segmented library. When virtual memory space is allocated, enough space is reserved to hold the largest size allowed for the data segment. In the case of a processes' data stack, this is the value of MAXDATA that was specified when the program was PREP'ed. The amount of virtual memory required is highly dependent on the users' applications as well as the number of users. Values of 4K and 8K are usually sufficient but large numbers of data segments or large data segment sizes may require more space. The virtual memory allocated to system domain volumes may be changed on any restart except a WARM-START. Logical Device 1 may have memory size changed only during RELOAD.

29. DIRECTORY

Appendix E provides a formula for estimating the amount of directory space required. Like virtual memory, the directory size can only be changed on a reload so it's advisable to err on the generous side.

30. MAX # OF SPOOLFILE KILOSECTORS

This parameter allows a system manager to specify the maximum amount of disc space that may be used by the spoolers. If this amount of space is exceeded, all spooling queues will be shut until they are opened by the operator. Many sites may choose to set this value large enough so that the effective limit on disc space is the amount of free space in the device class SPOOL.

31. # OF SECTORS PER SPOOLFILE EXTENT

This parameter specifies the increment that will be used for allocating spool file disc space. Since a spool file may have only 32 extents, it also specifies the maximum size of any single spool file. If large spool files are required, such as for year end reports, this value may be made quite large; there is a potential problem, though, since the system may be unable to find a large enough contiguous area of disc space to allocate an extent if the disc space is highly fragmented. This would make it impossible to do any spooling. An alternative to generating large output spool files is to periodically close the output file and open a new one. A large report program might start a new output file every 200 pages. While this would require gathering several files for the complete report, it has the advantage of allowing the first portion of the report to begin printing while the program is still running. Spool files use a special type of variable length records with trailing blanks stripped. An extent size of 384 sectors will allow an output file of approximately 25000 print lines.

32. MAX CODE SEGMENT SIZE
33. MAX STACK SIZE
34. MAX EXTRA DATA SEGMENT SIZE

These three parameters are used to limit the amount of resources that can be used by any one process. There sizes are application dependent.

The recommended max code segment size is between 4K and 8K words. This configuration size will restrict the programmer to segment sizes which can be most efficiently managed by the system.

The recommended configuration value for the maximum stack size is 31232, the maximum size allowed by MPE.

The recommended extra data segment size is 8K words.

35. MAX # OF CODE SEGMENT PER PROCESS

36. MAX # EXTRA DATA SEGMENTS PER PROCESS

These parameters are used to limit the amount of resources that can be used by any given process. If the system's users are conscientious about using the minimum resources required for a given task, these parameters may be set to the maximum allowed, thereby giving the users the maximum freedom. Otherwise, limits may be imposed. It is important to note, however, that these are system-wide limits that apply to all users.

37. STANDARD STACK SIZE

This is the default stack size used if the; STACK=parameter is not included in the :PREP and :RUN command. Since the amount of stack space needed is highly application dependent, users should be encouraged to tune their applications and use an explicit STACK declaration instead of relying on a default value.

Suggested Configuration Values for a 16 User System

	PARAMETER	SUGGESTED VALUE FOR 16 USERS	ENTRY SIZE IN WORDS	MEMORY RESIDENT
1.	MAX # OF OPEN SPOOLFILES	20	8	YES
2.	CST	192	4	YES
3.	EXTENDED CST	128-256	4	YES
4.	DST	192-320	4	YES
5.	PCB	96-192	16	YES
6.	DISC REQUEST TABLE	32-96	16	YES
7.	I/O QUEUE	64	11	YES
8.	TERMINAL BUFFERS PER PORT	*[1]	16 (Series II/III) 32 (Series 30/33/ 40/44/64)	YES
9.	SYSTEM BUFFERS	8-25	129	YES
10.	SWAP TABLE	*[2]	5	YES
11.	PRIMARY MESSAGE TABLE	25	5	YES
12.	SECONDARY MESSAGE TABLE	25	5	YES
13.	SPECIAL REQUEST TABLE	25	5	YES
14.	ICS	768	1	YES
15.	UCOP REQUEST QUEUE	32	2	NO
16.	TIMER REQUEST LIST	32	4	YES
17.	BREAKPOINT TABLE	32	Variable[3]	NO
18.	# USER LOGGING PROCESSES	3		
19.	# USERS PER PROCESS	16		
20.	# OF RINS	48	2	NO
21.	# OF GLOBAL RINS	16	12	NO
22.	# OF SECONDS TO LOGON	120		
23.	MAX # OF CONCURRENT SESSIONS	*[4]	1	YES
24.	MAX # OF CONCURRENT JOBS	*[5]	1	YES
25.	DEFAULT JOB CPU TIME LIMIT	0		
26.	LOG FILE RECORD SIZE	2		
27.	LOG FILE SIZE	1023		
28.	VIRTUAL MEMORY	4000-8000		
29.	DIRECTORY	768		
30.	MAX # OF SPOOLFILES KILOSECTORS	128		
31.	# OF SECTORS PER SPOOLFILE EXTENT	384		
32.	MAX CODE SEGMENT SIZE	8192		
33.	MAX STACK SIZE	31232		
34.	MAX EXTRA DATA SEGMENT SIZE	8192		
35.	MAX #OF CODE SEG/PROCESS	63		
36.	MAX # OF EXTRA DATA SEG/PROCESS	4		
37.	STANDARD STACK SIZE	1200-2000		

[1] Depends on #of concurrent terminals users and if block (page) mode is used.

[2] 5 times maximum number of PCB entries in the configured system.

[3] Variable in length, 5-12 words.

[4] #of available ports and #of PCB entries.

[5] Depends on system application.

LISTING FORMATS

APPENDIX D

:LISTACCT FORMAT

The format of the standard entry for each account is shown in figure D-1. In this format,

- ANAME and APASS are eight-character names, right-padded with blanks.
- The double-word numeric quantities are double-word integers, with %17777777777 representing "unlimited."
- ACAP can be decoded by reference to figure D-1 or to the WHO intrinsic, described in the *MPE Intrinsics Reference Manual.*
- MAX JOB PRIORITY is a numerical quantity.

The listing appears as an octal dump of the account entry or entries. Each entry is headed by the character *A* (for account), an equals sign, and the account name. Both user and system information are given in the listing. The user information in the entry is decoded as shown in table D-1.

A translation of those bytes that contain alphanumeric ASCII characters appears to the right of each line in the octal dump. A sample listing is shown in figure D-2.

Table D-1. User Information in Account Entry

WORDS	CONTENT
0-3	Account name.
6-7	Capability in same format returned by WHO intrinsic, described in the *MPE Intrinsics Reference Manual*.
8-9	Local attributes.
10-13	Password.
14-15	Permanent file space usage count (in sectors).
16-17	Permanent file space limit (in sectors).
18-19	Central processor time usage count (in seconds).
20-21	Central processor time limit (in seconds).
22-23	Connect-time count (in minutes).
24-25	Connect-time limit (in minutes).
26	Purge and account-security flags.
27	Maximum job/session priority (numerical).

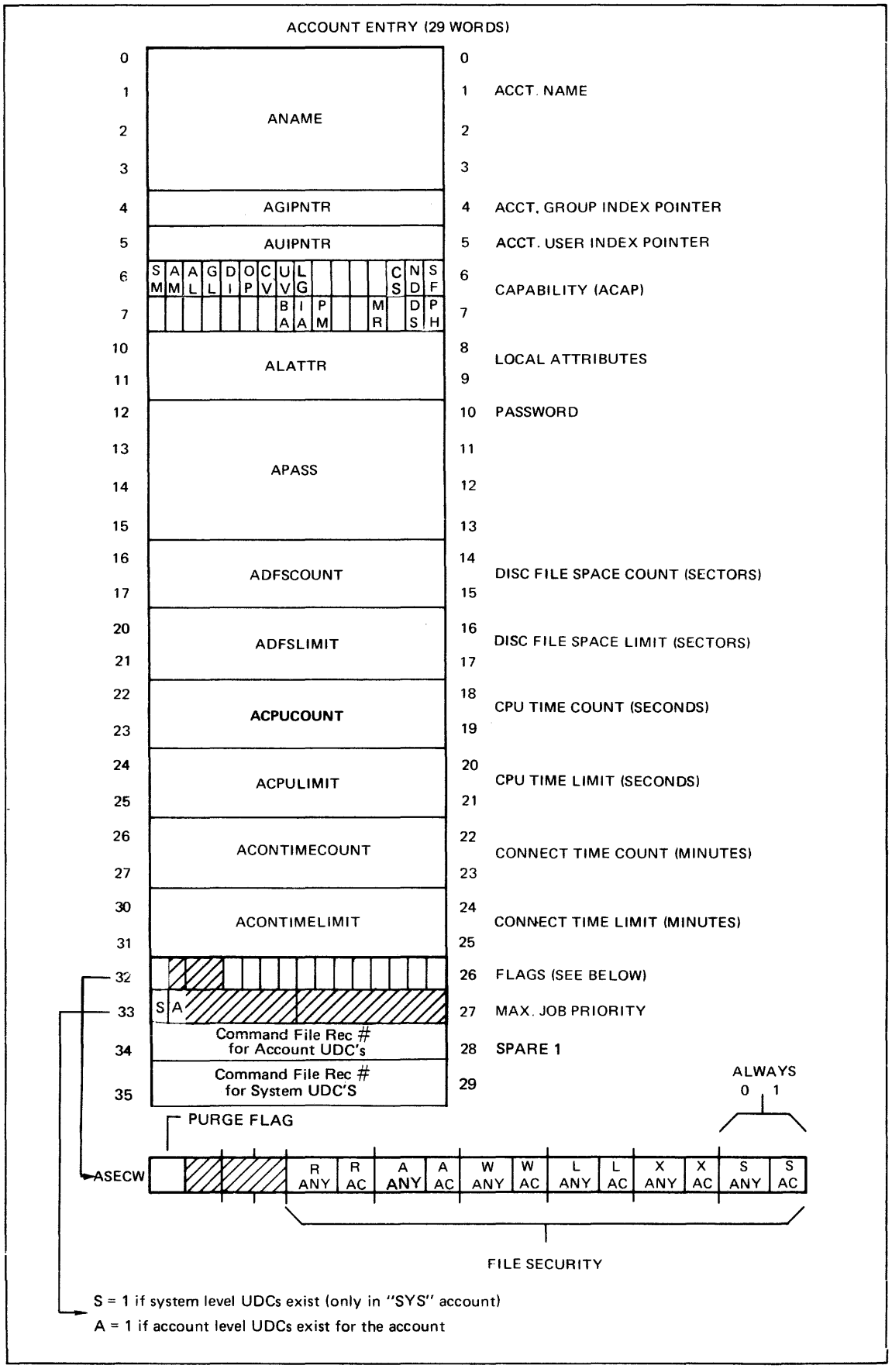


Figure D-1. Account Entry Format

```
:LISTACCT ↵
A  =  ACCT
040503 041524 020040 020040 000006 000007 070003 000600 ACCT........P...
000000 000000 020040 020040 020040 020040 000000 003323 ................
077777 177777 000000 000002 077777 177777 000000 000041 ................
077777 177777 002525 000226 000006 000000               .....U......
A  =  BTSS
041124 051523 020040 020040 000015 000016 070003 000600 BTSS........P...
000000 000000 020040 020040 020040 020040 000000 000103 ................C
077777 177777 000000 000556 077777 177777 000000 000065 ........N........5
077777 177777 002525 000226 000000 000000               .....U......
A  =  CANFIELD
041501 047106 044505 046104 000021 000022 176007 000713 CANFIELD........
000000 000000 020040 020040 020040 020040 000000 013741 ................
077777 177777 000000 057052 077777 177777 000000 033204 ..............6.
077777 177777 002525 000144 000030 000000               .....U.D....
A  =  CLIFTON
041514 044506 052117 047040 000025 000026 177007 001773 CLIFTON.........
000000 000000 041501 051040 020040 020040 000000 001312 ....CAR.........
077777 177777 000000 003344 077777 177777 000000 003462 ..............2
077777 177777 002525 000226 000035 000000               .....U......
A  =  CROLEY
041522 047514 042531 020040 000031 000032 177007 001773 CROLEY..........
000000 000000 044101 051122 042514 046040 000000 003036 ....HARPELL.....
077777 177777 000000 000345 077777 177777 000000 000461 ................1
077777 177777 002525 000226 000042 000000               .....U......
A  =  DATAMGT
042101 052101 046507 052040 000035 000036 177407 000713 DATAMGT.........
000000 000000 020040 020040 020040 020040 000000 002535 ................
077777 177777 000000 030135 077777 177777 000000 012547 .......0........G
077777 177777 002525 000226 000000 000000               .....U......
```

Figure D-2. Sample :LISTACCT Listing

:LISTGROUP FORMAT

The format of the entry for each group is shown in figure D-3

- GNAME and GPASS are eight-character names, right-padded with blanks.
- The double-word numeric quantities are double-word integers, with %17777777777 representing "unlimited."
- GCAPABILITY is in the same format as capability-class attributes (Word 2 of CAP), as returned by the WHO intrinsic.

The group listing appears as an octal dump of the group entry or entries. Each entry is headed by the character *G* (for group), an equals sign, and the group name. Both user and system information are given in the listing. The user information in the entry is decoded as shown in table D-2.

A translation of those bytes that contain alphanumeric ASCII characters appears to the right of each line in the octal dump.

Table D-2. User Information in Group Entry

WORDS	CONTENT
0-3	Group name.
5-8	Password.
9-10	Permanent file space usage count (in sectors).
11-12	Permanent file space limit (in sectors).
13 14	Central-processor time usage count (in seconds).
15-16	Central-processor time limit (in seconds).
17-18	Connect-time count (in minutes).
19-20	Connect-time limit (in minutes).
21-22	Purge and group-security flags.
23	Capability-class attributes (in the same bit-format returned by the WHO intrinsic, described in the *MPE Intrinsics Reference Manual).*
24	Group directory base linkage.
25	Group volume set definition index.
26-37	Group home volume set name.
38	GSAVEFIPNTR
39	GMOUNTREFCNTR
40	GSPARE

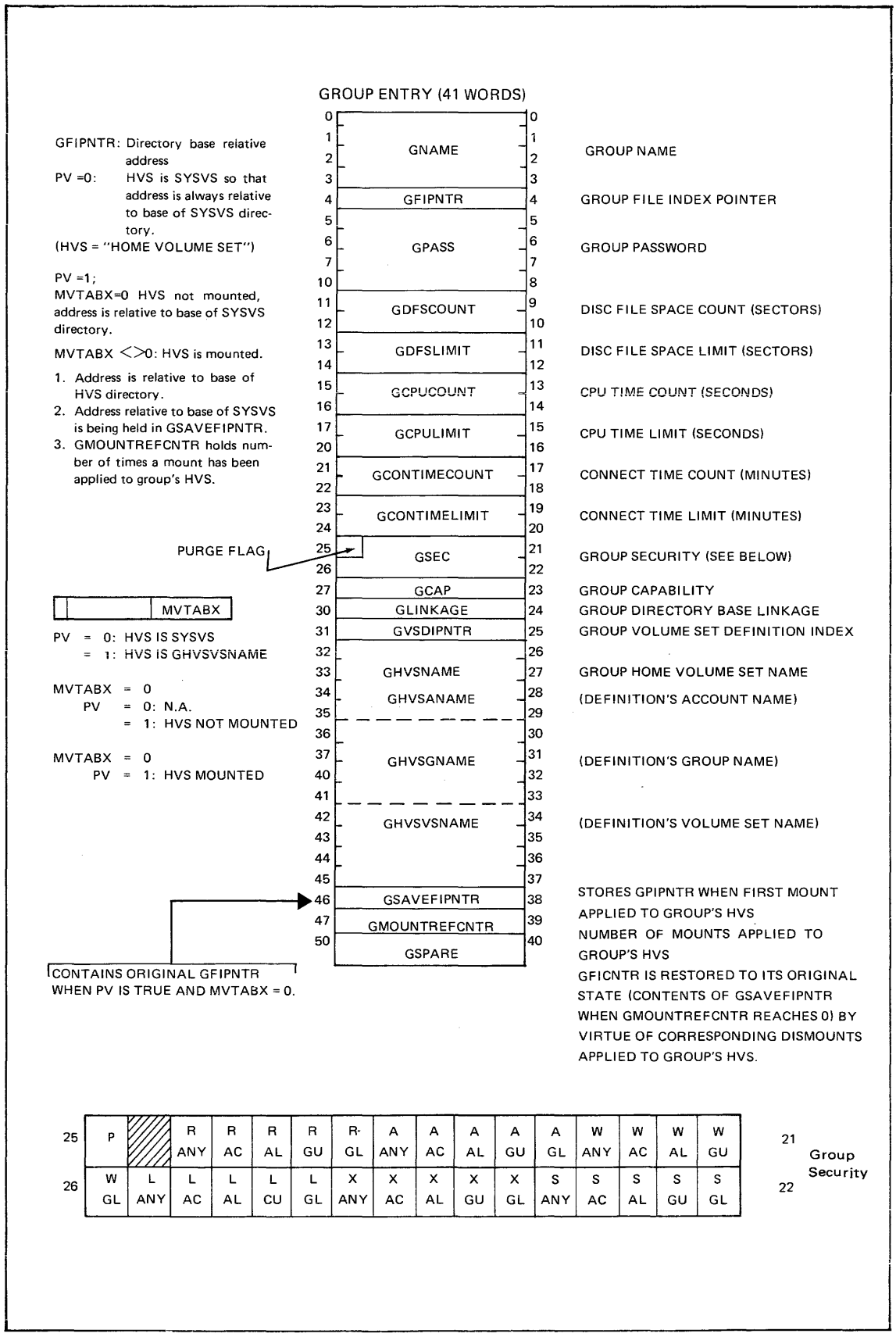
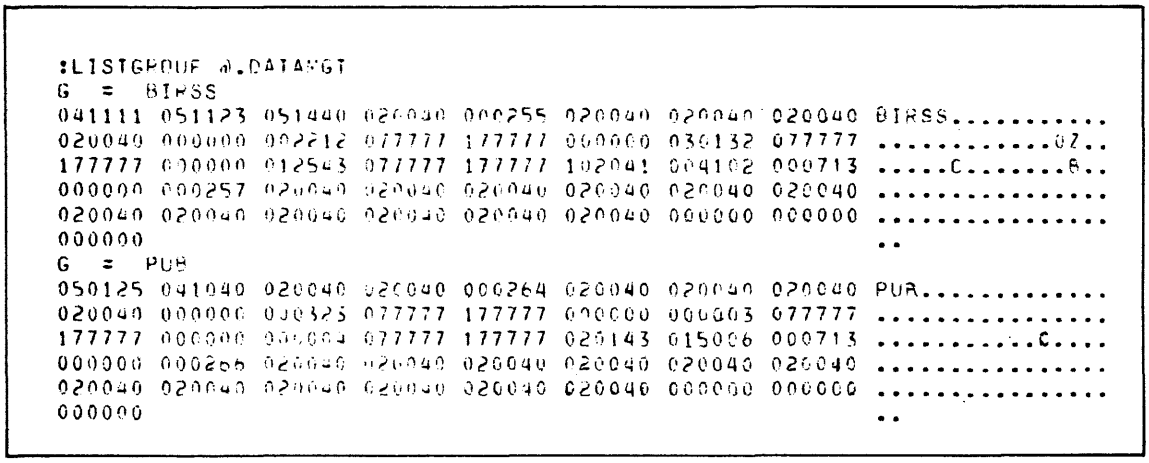


Figure D-3. Group Entry Format



```
:LISTGROUP @.DATAMGT
G  =  BIRSS
041111 051123 051440 020040 000255 020040 020040 020040 BIRSS.....
020040 000000 002212 077777 177777 000000 030132 077777 .....0Z..
177777 000000 012543 077777 177777 102041 004102 000713 .....C.....B..
000000 000257 020040 020040 020040 020040 020040 020040 ..............
020040 020040 020040 020040 020040 020040 000000 000000 ..............
000000                                                  ..
G  =  PUB
050125 041040 020040 020040 000264 020040 020040 020040 PUR.....
020040 000000 000325 077777 177777 000000 000003 077777 ..............
177777 000000 000004 077777 177777 020143 015006 000713 .....C....
000000 000266 020040 020040 020040 020040 020040 020040 ..............
020040 020040 020040 020040 020040 020040 000000 000000 ..............
000000                                                  ..
```

Figure D-4. Sample :LISTGROUP Listing

:LISTUSER FORMAT

The format of the entry defining each user is shown in figure D-5.

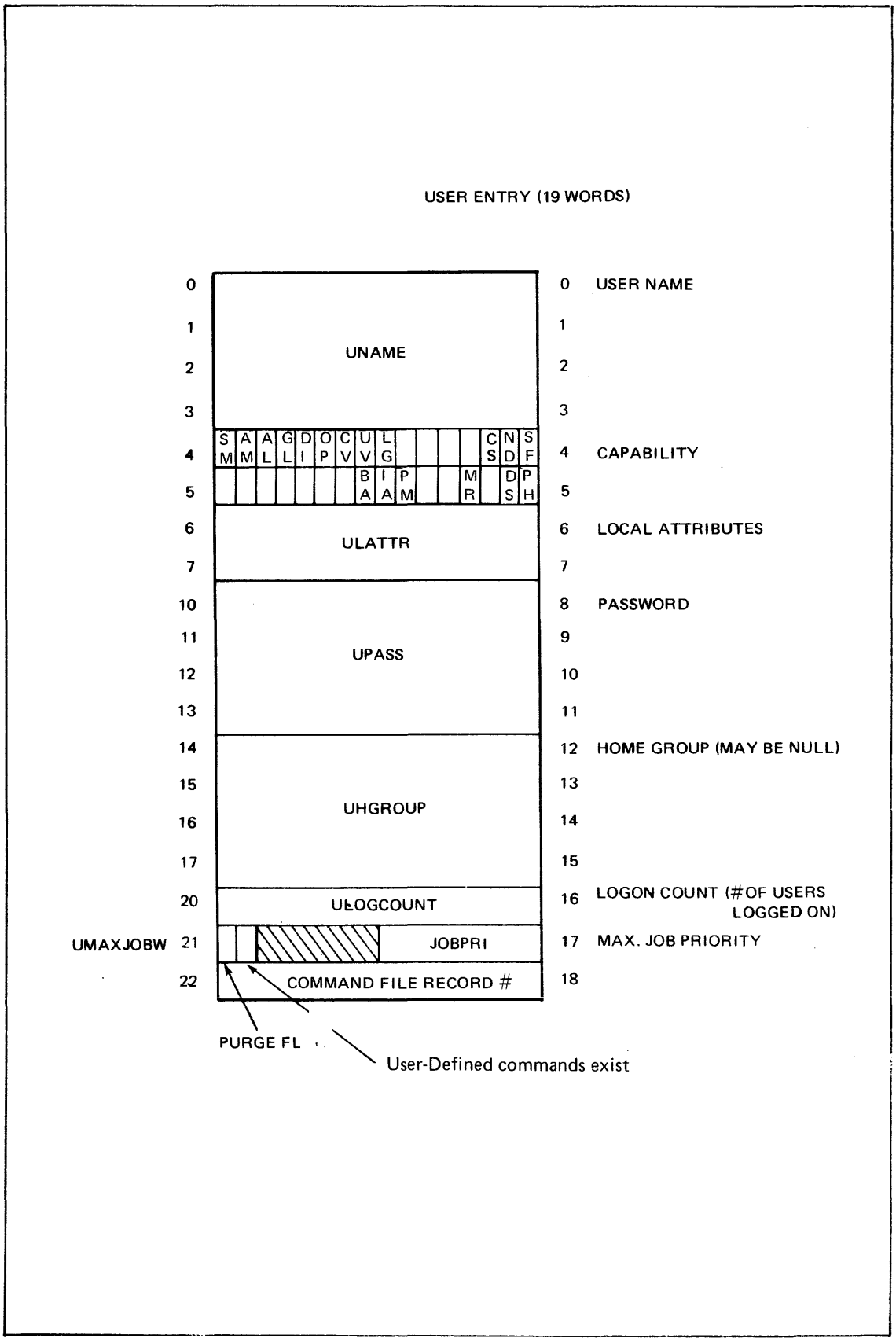
- UNAME, UPASS and UHGROUP are eight-character names, right-padded with blanks.
- UCAP can be decoded by reference to the WHO intrinsic.
- JOBPRI is a numerical quantity.

The user listing appears as an octal dump of the user entry or entries. Each entry is headed by the character *U* (for user), an equals sign, and the user's name. Both user and system information are given in the listing. The user information in the entry is decoded as shown in table D-3.

A translation of those bytes that contain alphanumeric ASCII characters appears to the right of each line in the octal dump. A sample listing is shown in figure D-6; in this example, a *System/Account Manager User wants to list the attributes of all users in the account named LANG.*

Table D-3. User Information in User Entry

WORDS	CONTENT
0-3	User name.
4-5	Capability (in the same bit-format returned by the WHO intrinsic, described in the *MPE Intrinsics Reference Manual*).
6-7	Local attributes.
8-11	Password.
12-15	Home group.
16	Number of users logged-on.
17	Maximum job priority (numerical); 150 (226_8) is roughly equivalent to CS subqueue.



Figur D-5. User Entry Format

```
:LISTUSER @.LANG

U=  BAUSEK
041101 052523 042513 020040 031003 000700 000000 000000 BAUSEK..2.......
020040 020040 020040 020040 041517 041117 046040 020040 ........COBOL...
000000 000226 000000                                    ......

U=  COUCH
041517 052503 044040 020040 001003 000601 000000 000000 COUCH...........
020040 020040 020040 020040 041101 051511 041517 046520 ........BASICOMP
000000 000226 000000                                    ......

U=  SMITH
043522 042505 047040 020040 001003 000601 000000 000000 SMITH...........
020040 020040 020040 020040 041101 051511 041440 020040 ........BASIC...
000000 000226 000000                                    ......

U=  MANAGER
046501 047101 043505 051040 071003 000701 000000 000000 MANAGER.R.......
046117 041467 030462 030040 050125 041040 020040 020040 LOC7120.PUB.....
000000 000036 000000                                    ......

U=  MARTIN
046501 051124 044516 020040 001003 000601 000000 000000 MARTIN..........
020040 020040 020040 020040 041101 051511 041440 020040 ........BASIC...
000000 000226 000000                                    ......

U=  JONES
051510 044520 046501 047040 001003 000601 000000 000000 JONES...........
020040 020040 020040 020040 041101 051511 041440 020040 ........BASIC...
000000 000226 000000                                    ......
```

Figure D-6. Sample :LISTUSER Listing

:SHOWQ FORMAT

The format for the information displayed by the :SHOWQ command is shown in figure D-7. The information displayed is divided into two groupings, each headed by Q, PIN, and JOBNUM.

Processes in the right group are listed in high-to-low order. These processes currently, or will imminently, require the CPU in order to continue. The CPU will automatically be given to the highest priority process.

Processes in the left group are listed in numerical order, and are waiting on longer term events. They will again contend for the CPU when those events occur.

```
:SHOWQ

 DORMANT                    WAITING                    RUNNING
Q   PIN     JOBNUM         Q    PIN     JOBNUM        Q    PIN     JOBNUM

L       1                                             C    M18     #S483
L       2
L       3
L       4
L       5
L       6
L       7
L       8
L       9
L      10
C     U12   #S454
C     M13   #S478
L      15
C     U22   #S480
D     M32   #J8
L      74
C     U75   #S446
C     M77   #S480
C     M79   #S446
C     M81   #S454
L      96
C    M119   #S324
C    M124   #S474
D    U138   #J8
C    U152   #S446

CQ MINQUANTUM=0, MAXQUANTUM=300, BASEPRI=152, LIMITPRI=200
DQ MINQUANTUM=1000, MAXQUANTUM=1000, BASEPRI=202, LIMITPRI=238
EQ MINQUANTUM=1000, MAXQUANTUM=1000, BASEPRI=240, LIMITPRI=253
MINIMUM CLOCK CYCLE=1000
```

Figure D-7. :SHOWQ List Format

After displaying the process queues, :SHOWQ prints the scheduling parameters currently in effect, and the minimum algorithm cycle value (minclockcycle).

NOTE

On occasion, a process will apear in both groups. This indicates that the process was being moved to the right during the execution of :SHOWQ.

The following information is displayed for each process.

$$\left\{\begin{matrix}L\\C\\D\\E\end{matrix}\right\}\ \left[\left\{\begin{matrix}M\\U\end{matrix}\right\}\right]\quad <pin>\left[\#\left\{\begin{matrix}J\\S\end{matrix}\right\}\ <jobnum>\right]$$

where

L is a linearly scheduled process on the AS or BS subqueues, or Master queue.

C is a circularly scheduled process on the CS subqueue.

D is a circularly scheduled process on the DS subqueue.

E is a circularly scheduled process on the ES subqueue.

M is a job/session main process.

U is a user process.

pin is the process identification number for this process.

J is the process executing in a batch job.

S is the process being executed from a session.

jobnum is the job or session number.

NOTE

On occasion, there will be an incomplete entry for a process. This indicates that MPE is currently building this entry in the Process Control Block Table.

DISC SPACE UTILIZATION

APPENDIX E

The system disc is used primarily for four system requirements: MPE virtual memory, the disc directory, MPE files, and information needed during cold loading. Any space remaining is available for user files.

Sectors 0 through 399 are reserved for the disc volume label, defective tracks table or defective sectors table, and bootstrap input/output program.

The disc free space map, the directory, and virtual memory are located at variable positions on the disc. The sizes of the directory and virtual memory are defined by parameters specified at system configuration time.

Other portions of the disc are used for the system files, message catalog and tables used by INITIAL and INITIAL program segments. (This includes the SL file containing the Scientific, Compiler and the software run time libraries for the languages).

The following formula should be used to determine the approximate numbers of sectors needed for the directory:

SECTORS = 6+6* A+(5.4*G)+(.15*U)+(.05*F)+(.5*VS)+(.5*VC)

where

A = Total number of accounts in system.
G = Total number of groups in system.
U = Total number of users in system.
F = Total number of files in system.
VS = Total number of volume sets in the system.
VC = Total number of volume classes in the system.

Maximum values permitted are:

Accounts = 632
Groups per account = 96
Files per group = 1385
Users per account = 200
Volume sets per group = 32
Volume classes per group = 32

Virtual memory is used only for the swapping of data segments. It is allocated in 4-sector blocks. Approximately 250 sectors are needed for system data segments. For each user on the system, the following amount of virtual memory is needed:

32	sectors for Command Interpreter stack
8	sectors (approximately) for all unbuffered files, depending on buffer size
4	sectors per open buffered file
16	sectors for system area in the user's stack plus 4 sectors per 512 words in the DL/Z area of the stack.

In addition, while a program is being loaded, 40 sectors of virtual memory are needed for the Loader.

To determine the amount of virtual memory needed, estimate the average number of concurrent users, average stack size, average number of buffered files open per user, and number of users who will be loading programs at the same time; then use these figures in conjunction with the above values.

MISCELLANEOUS FORMATS

APPENDIX F

Table F-1. :STORE Tape Format

ITEM NO.	ITEM
1	End-of-File (EOF) Mark
2	EOF Mark
3	Header label (40 words), used as follows: **Words** — **Contents** 0-13 "STORE/RESTORE LABEL-HP/3000." 14-15 MPE III Identification. (Identifies modified format of :STORE tape). 16 If true, indicates that first file on volume is continued from previous volume. 17 Checksum for verifying validity of header (XOR of all 40 words of the header). 18 Index into STORE directory of first file on volume (could be a continuation of a file from previous volume). 19-22 Reserved by MPE. 23 Reel number. 24 Bits (0:7) = last 2 digits of year; (7:9) = Julian date } Date of creation. 25 Bits (0:8) = hours; (8:8) = minutes 26 Bits (0:8) = seconds; (8:8) = tenth-of-seconds } Time of creation. 27 Block size in words. "0" means 1024 words.
4	EOF Mark
5	Tape directory — Consists of 12-word records, blocked according to the tape block size specified. (The last record may be shorter.) There is one entry for each file on the tape, and the entries are ordered the same as the files. The 12-word entry is: **Word** — **Contents** 0-3 File name 4-7 Group name 8-11 Account name
6	EOF Mark
7	First file. The data is blocked according to the block size specified in the store tape :FILE command. (The last record may be shorter, but will always be a multiple of 256 words.) For fixed-length and undefined-length record files, only data up to the end-of-file is dumped; intervening zero-length extents are not dumped. For variable-length record files, only allocated extents are dumped.

Table F-1. :STORE Tape Format (continued)

ITEM NO.	ITEM
8	EOF Mark
9	Second File
10	EOF Mark : :
11	Last File
12	EOF Mark
13	Trailer Label (40 words). Identical to header label (Item 3) except that Words 21 and 22 are used as follows: **Word** **Use** = 0 means that preceding file continues on next tape volume 21 = 1 means that preceding file ended with preceding EOF mark = H word 22 = 0
14	EOF Mark
15	EOF Mark
16	EOF Mark

:STORE tapes may have multiple reels. If end-of-tape (EOT) is detected during a write data operation, a file mark is written followed by Items 13 to 16 above, with word 21 of the trailer label set to 1 if this was the last recrod of the file and 0 otherwise. If EOT is detected on a write file mark operation, Items 13 to 16 are written, with word 21 set to 1 and word 22 set to 1 if this is the last file on the tape, and 0 otherwise. Reels subsequent to Reel 1 have the following format:

1. Header label.
2. EOF mark.
3. Remainder of preceding file or next file.
4. EOF mark.
5. Next file; the rest of the tape is written in the same format as the first reel.

Each file on the tape is written in blocks which you may specify in the related :FILE equation to be from 256 to 4096 words long, in increments of 256 words. The default value is 4096 words for non-programmatic calls to :STORE, and 1024 words for programmatic calls.

Table F-2. Labeled :STORE Tape Format

Note: Sysdump to a labeled tape will not recover the MPE portion of the tape.

ITEM NO.	ITEM	
1		Volume 1 Label (40 words)
2		Header 1 Label (40 words)
3		Header 2 Label (40 words)
4		UHL (User Header Label)
	WORDS	**CONTENTS**
	0-13	"STORE/RESTORE LABEL – HP/3000"
	14-15	MPE Identification. (Identifies modified format of :STORE tape).
	16	If true, indicates that first file on volume is continued from previous volume.
	17	Checksum for verifying validity of header (XOR of all 40 words of the header)
	18	Index into STORE directory of first file on volume (could be a continuation of a file from a previous volume)
	19-21	**Reserved by MPE**
	22	**Block size in words 'O' means 1024 words.**
	23	Reel number
	24	Bits (0:7) = last 2 digits of year Bits (7:9) = Julian date } Date of Creation
	25	Bits (0:8) = hours
		(8:8) = minutes
	26	(8:8) = seconds } Time of Creation
		(8:8) = tenths
5		Tape Mark
6		Tape Directory – Consists of 12-word records, blocked according to the tape block size specified. (The last record may be shorter.) There is one entry for each file on the tape, and the entries are ordered the same as the files. The 12-word entry is:
	WORD	**CONTENTS**
	0-3	File name
	4-7	Group name
	8-11	Account name
7	Tape Mark	
8	EOF 1	
9	EOF 2	

Table F-2. Labeled :STORE Tape Format (continued)

WORD	CONTENTS
10	
11	Header 1
12	Header 2
13	Tape Mark
14	File #1. The data is blocked according to the block size specified in the labeled store tape :FILE command. (The last record may be shorter, but it will always be a multiple of 256 words.) For fixed-length and undefined-length record files, only data up to the end-of-file is dumped; intervening zero-length extents are not dumped. For variable-length record files, only allocated extents are dumped.
15	Tape Mark
16	EOF 1
17	EOF 2
18	Tape Mark
19	Header 1
20	Header 2
21	Tape Mark
22	File #2
23	Tape Mark
24	EOV 1
25	EOV 2
26	Tape Mark
27	Tape Mark

Labeled :STORE tapes may have multiple reels. When the file spans more than one volume, EOV is written instead of EOF (items 23 and 24). Reels subsequent to Reel 1 have the following format:

ITEM NO.	ITEM
1	Volume 1
2	Header 1
3	Header 2
4	UHL (same format as Reel 1)
5	Tape Mark
6	Remainder of preceding file
7	Tape Mark
8	EOF 1
9	EOF 2
10	Tape Mark

Table F-2. Labeled :STORE Tape Format (Continued)

ITEM NO.	ITEM
11	Header 1
12	Header 2
13	Tape Mark
14	Tape Directory
15	Tape Mark
16	EOF 1
17	EOF 2
18	Tape Mark
19	Header 1
20	Header 2
21	Tape Mark
23	File #3
24	Tape Mark
25	EOF 1
26	EOF 2
27	Tape Mark
28	Tape Mark

Indicates end of tape

Table F-3. :SYSDUMP Tape Format

ITEM NO.	ITEM
1	Bootstrap – record. 32 words long, read by the cold load microcode and containing a six-word SIO program which reads in the next record. This record exists only for Series II/III systems.
2	I/O program – A SIO input/output program to read the records defined in items 3 through 22. These records exist only for Series II/III systems.
3	Bootstrap – a record. 128 words long, read by the cold load microcode and containing an HPIB channel program which reads in the next record.
4	I/O program – An HPIB channel program to read the records defined in items 5 through 24.
5	ICS and entry point (32 words) – The Interrupt Control Stack (ICS) containing the initial values for DB, DL, Z, Q and S.
6	Low-core (12 words). Initial values for absolute locations 0-11.
7	CSTAB (Blocked 1024 words per record) COmmunications Subsystem Table.
8	Driver Table (Blocked 1024 words per record). Driver name for each DRT and core-resident flag, as updated by configuration changes.
9	Logical-Physical Device Table (Blocked 1024 words per record) – As updated by input/output configuration changes.
10	LDT, Logical Device Table (Blocked 1024 words per second) – As updated by input/output configuration changes.
11	Device Class Table (Blocked 1024 words per record) – As updated by input/output configuration changes.
12	LDTX, Logical Device Table Extension (Blocked 1024 words per record).
13	Volume Table (Blocked 1024 words per record) – Modified volume table.
14	Old Volume Table (Blocked 1024 words per record) – Volume table as it was before any changes made to it on this :SYSDUMP. This table is included only if the user files are being dumped also.
15	Disc Cold Load Information Table (256 words) – Sectors 18 and 19 of the system disc. This table is included only if the user files are being dumped also.
16	Core Size Related Configuration Table (1024 words) – 128 words for each of the 8 configurable core sizes.
17	Configuration Table (128 words) – Configuration information not related to core size.
18	CS Drivers (128 words) – Additional CS drivers.
19	CSDEF (256 words).
20	Internal Interrupts (64 words) – Halt instruction for each of the internal interrupts.

Table F-3. :SYSDUMP Tape Format (continued)

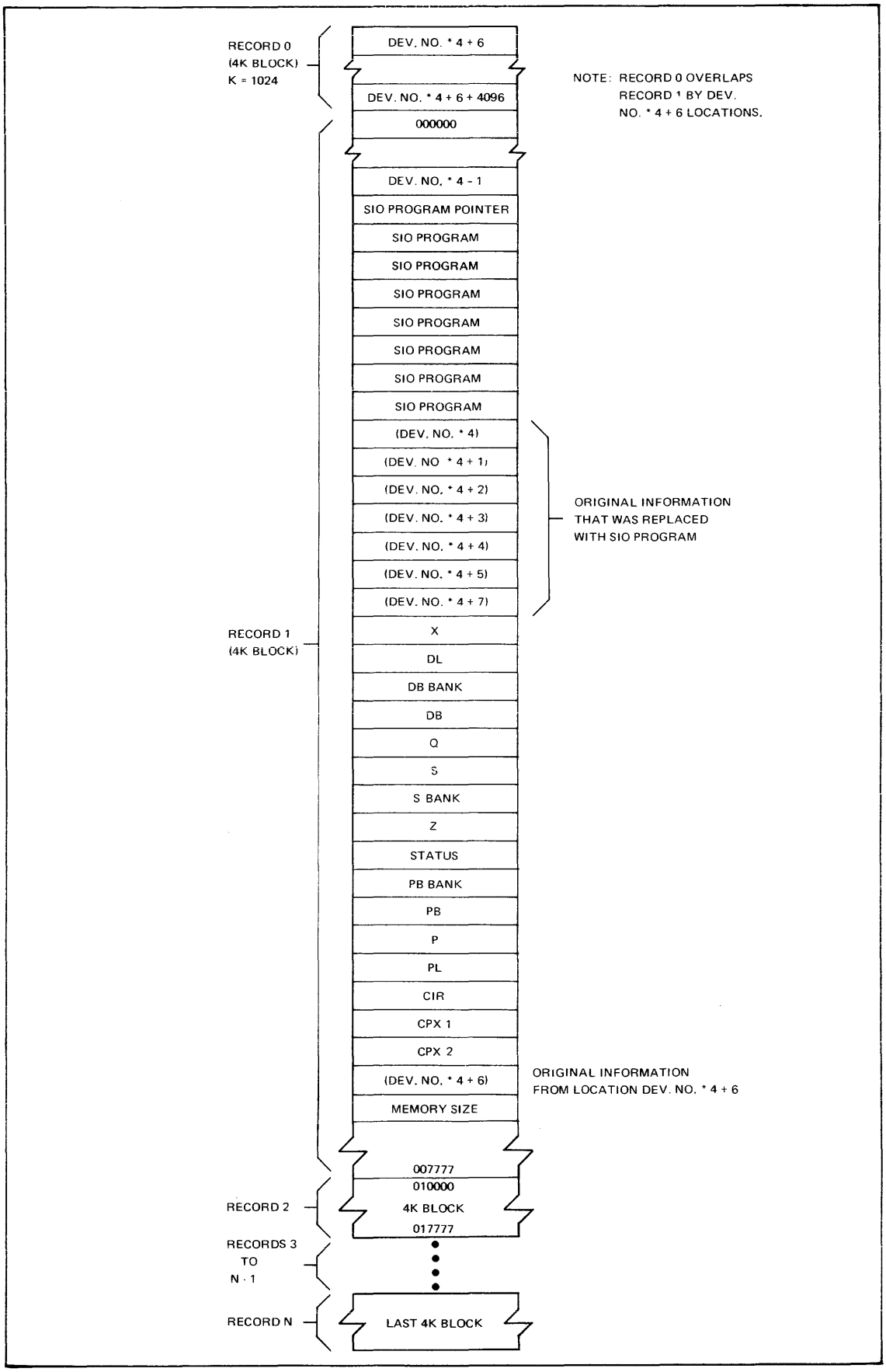
ITEM NO.	ITEM
21	Temporary CST (128 words) — Code segment table (CST) used for INITIAL's segments while INITIAL is running.
22	INITIAL's DB area (Blocked 1024 words per record) — Initialized DB area for INITIAL.
23	INITIAL marker (12 words) — Initial stack marker for INITIAL program.
24	INITIAL Segments from %17 to 6 (Blocked 1024 words per record for each segment) — Those segments of INITIAL which are read in by the cold load input/output program.
25	INITIAL Segments 5-0 (Blocked 1024 words per record for each segment) — Those segments of INITIAL which are not in core when the machine halts following cold load.
26	Header Record (20 words) and Message Catalog (Blocked 1024 words per record) — Table of MPE messages.
27	RIN Table (Blocked 1024 words per record) — If user files are being dumped this table is included and shows the current RIN table.
28	Header Record (20 words) and Directory (Blocked 1024 words per record) — Included only if user files are being dumped.
29	End-of-File mark.
30	System files — Each allocated extent of each of the system files is dumped, blocked 1024 words per record. Files always dumped if they are on the disc (all in PUB SYS) are: SL, HIOTAPEØ, HIOLPRTØ, HIOLPRT1, HIOFLOPØ, HIOMOSC1, IOINPØ, LOG, DEVREC, PROGEN, UCOP, IOCDPNØ, LOAD, DISPATCH, SEGDUR, SEGPROC, SYSDUMP, INITIAL, CONFDATA, PFAIL, MAKECAT, MENLOGP, CATALOG, PVPROC, PVINIT, MPECHECK, IODSØ, CSDUMMY, HIOLPRT2, HIOPRTØ, HIOTAPE1. These programs are dumped only if the system is a Series II/III: IOTAPEØ, IOLPRTØ, IOTERMØ, IOMDISCØ, IOFDISCØ, IOCDRDØ, CSHBSCØ, IOREMØ, IOPTRDØ, IOPTPNØ, IOPLOTØ, IOPRPNØ, IOMDISC1, CSSBSCØ. These programs are dumped only if the system is a Series 30/33/40/44/64: SDFLOAD, SDFCOM, SDFCHECK, SDFGEN, HIOTERM0. In addition, any input/output drivers defined in the configuration but not in the above list are dumped.
31	End-of-File Mark.
32	If any user files were to be dumped, the rest of the tape is identical to the :STORE tape format described in table F-1, (beginning with Item 3, the header label). The presence of the two file marks written as Items 27 and 29 above, results in :STORE and :SYSDUMP tape compatibility for use by :RESTORE. If no user files were to be dumped, the remainder of the tape is the same as a :STORE tape with no files, consisting of a header label, two end-of-file marks, a trailer label, and three end-of-file marks.

Table F-4. Disc Volume Labels

When each disc is initialized, MPE writes a volume label in the first sector (Sector 0). This volume label is written in the following format:

WORDS	CONTENTS
0-5	For Series II/III, on the system disc, this field contains the bootstrap input/output program. On other discs, this field is filled with zeros. For Series 30/33/40/44/64, this field contains the string "SYSTEM DISC".
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).

Table F-5. System Dump Magnetic Tape Format



MPE MESSAGE SYSTEM

APPENDIX G

The MPE message system consists of a message catalog (CATALOG.PUB.SYS), the Help subsystem catalog (CICAT, containing descriptions of all MPE commands), and a program (MAKECAT) for managing these catalogs.

MESSAGE CATALOG

The message catalog (CATALOG.PUB.SYS) is a numbered Editor file containing message sets. The sets serve to break the catalog into manageable portions.

Messages in the catalog can be of any length and can contain up to five parameters. Continuation of a message is indicated by "%" or "&" at the end of a line. The symbol % indicates that the message is continued and that a carriage return, line feed should be performed. The & symbol indicates that the message is continued on the same line with no carriage return, line feed. Message numbers need not be contiguous, but must be in ascending order.

The position of parameters in a message is indicated by the symbol "!". Message sets are indicated by "$SET n", starting in column 1 (the rest of the line is a comment). Maximum value for n is 20. Comments can be inserted in the catalog by placing "$" in column 1. After processing by the MAKECAT program, the catalog file contains records of 80 bytes, blocked 16, with one extent. The format of the message catalog is as follows:

```
$SET 1  SYSTEM MESSAGES
1 LDEV #! IN USE BY FILE SYSTEM
2 LDEV #! IN USE BY DIAGNOSTICS
  .
  .
  .
$ MESSAGE 57 DELETED
  .
  .
  .
204 FILE COMMAND REQUIRES AT LEAST TWO PARAMETERS, INCLUDING
THE FORMAL NAME OF THE FILE (CIERR 204)
```

Table G-1. CATALOG.PUB.SYS

$SET 1 — SYSTEM MESSAGES (FORMER MESSAGE CATALOG)
$SET 2 — CIERROR MESSAGES
$SET 3 — MISCELLANEOUS ABORT MESSAGES
$SET 4 — PROGRAM ERROR ABORT MESSAGES
$SET 5 — INTRINSIC ABORT MESSAGES
$SET 6 — RUN-TIME ABORT MESSAGES
$SET 7 — CI GENERAL MESSAGES
$SET 8 — FILE SYSTEM ERROR MESSAGES
$SET 9 — LOADER ERROR MESSAGES
$SET 10 — CREATE ERROR MESSAGES
$SET 11 — ACTIVATE ERROR MESSAGES
$SET 12 — SUSPEND ERROR MESSAGES
$SET 13 — MYCOMMAND ERROR MESSAGES
$SET 14 — LOCKGLORIN ERROR MESSAGES
$SET 15 — PRIVATE VOLUME ERROR MESSAGES
$SET 16 — DS/3000 ERROR MESSAGES
$SET 17 — HELP ERROR MESSAGES
$SET 18 — *NOT USED*
$SET 19 — SERIAL DISC ERROR MESSAGES
$SET 20 — USER LOGGING

MAKECAT PROGRAM

The MAKECAT program is used to build CATALOG and CICAT. The program has the following entry points:

Beginning of program (no entry point) — Builds a new CATALOG, but CATALOG is a temporary file. Also renames any old temporary catalog CATXXXX. Thus the catalog can be verified without actually having to install it.

BUILD — Reads input file, installs this as a new catalog in the system, and moves directory into a data segment. This can be done while the system is running. (Must log on as MANAGER.SYS to use this entry point.)

DIR — Reads existing directory from CATALOG.PUB.SYS.

HELP — Reads input file (formal designator INPUT), builds new file with name HELPCAT.

MODIFYING CATALOG WITH MAKECAT

You must have System Manager capability to modify CATALOG. You do not need System Manager capability to add to CATALOG, or to add to or modify CICAT. The procedure is as follows:

1. Log on as System Manager.
2. Text the file (CATALOG.PUB.SYS) into the Editor.
3. Make the change or addition.
4. Keep the file under a new name and exit the Editor.
5. Enter the following file equation:

   :FILE INPUT=newname

6. Perform one of the following:

   a. Run MAKECAT,BUILD [Entry point]
   (If everything is satisfactory, MAKECAT prints "XX NEW CATALOG INSTALLED".)

   Purge old catalog CATnnnn, where nnnn is an archival number, CAT0001, CAT0002, etc.

   b. Perform a SYSDUMP. (SYSDUMP runs MAKECAT, replaces CATALOG.)

EXAMPLES:

1) A normal user creates his own message catalog, keeps it in file MYMSGS.

```
:FILE INPUT = MYMSGS
: RUN MAKECAT.PUB.SYS
:SAVE CATALOG
```

2) The system manager changes or adds a few messages to CATALOG.PUB.SYS., keeps it in file NEWCAT.

```
:FILE INPUT = NEWCAT
:RUN MAKECAT.PUB.SYS. BUILD
```

3) The system manager restores CATALOG.PUB.SYS from mag tape. This is restored only if the file CATALOG.PUB.SYS no longer exists on the system.

```
:RESTORE *T; CATALOG.PUB.SYS
:RUN MAKECAT.PUB.SYS, DIR
```

HELP SUBSYSTEM CATALOG (CICAT)

The Help subsystem catalog (CICAT) contains descriptions of all MPE commands. MPE commands in the catalog are indicated by \ENTRY; subparameters (parameters, operation, example) are indicated by \ITEM or \SUBITEM. \STOPHELP causes Help to skip over any

subsequent lines until \STARTHELP is encountered. \STARTHELP causes Help to start displaying again. \SUBSET, placed at beginning of file, causes MAKECAT to build a file that omits the lines between \STOPHELP\STARTHELP pairs. Consequently makes Help file smaller.

Thus, a typical command entry in the catalog is of the form:

```
\ENTRY=ABORT
   :ABORT

   Aborts current program or operation.

   SYNTAX

      :ABORT

   \ITEM=PARMS
   PARAMETERS

   None
   \ITEM=OPERATION
   OPERATION

   After you suspend a program or MPE command operation by pressing the BREAK key,
   the :ABORT command immediately terminates that program or operation. Note that
   the :ABORT command is available only from a session and only during a break. An
   :ABORT command results in the Job Control Word (JCW) being set to the SYSTEM 0
   state.

   \ITEM=EXAMPLE
   EXAMPLE

   To abort the current operation, enter:

      :ABORT
```

MODIFYING THE HELP SUBSYSTEM CATALOG WITH MAKECAT

To modify CICAT, perform the following:

1. Log on as System Manager.
2. Text the file (CICAT.PUB.SYS) into the Editor.
3. Make the change or addition.
4. Keep the file under the new name CATNEW.
5. Enter the following file equations:

```
:PURGE CICAT
:FILE INPUT=CATNEW
:FILE HELPCAT=CICAT
```

6. Run MAKECAT,HELP [Entry Point]

Note that the new CICAT file can be stored offline on tape or serial disc.

DISC PACK FORMATTING

APPENDIX H

MOVING-HEAD DISC PACK FORMATTING ON THE SERIES II/III

Moving-head disc packs can be formatted through the use of a stand-alone diagnostic called SLEUTH. This diagnostic is on a stand-alone magnetic tape. The diagnostic is known as PD411A.

SLEUTH is designed for many types of device analysis. In this respect, we are only interested in how an Operator/System Supervisor will format disc packs. This information is also documented in the *SLEUTH Diagnostic Manual.*

OPERATING INSTRUCTIONS

STEP	PROCEDURE
1	Mount the cold-loadable magnetic tape containing SLEUTH on a magnetic tape drive. Set the SWITCH REGISTER on the control panel to octal 003006. Ensure that unit 0 is selected for the drive and it is on-line.
2	While pressing (and holding) the ENABLE switch, press the LOAD switch. This action causes the first portion of the master diagnostic program to be read into main memory. The system will then halt.
3	Set the SWITCH REGISTER to the number which indicates the physical position of SLEUTH on the diagnostic magnetic tape. This number must be obtained from your Hewlett-Packard Customer Engineer prior to execution. Press RUN switch. The system will search the magnetic tape for the SLEUTH program. It will be read into main memory when found. Tape unit will rewind.
4	When tape is back at load point, the diagnostic is now in execution. Next, depress the carriage return key on the system console so that the hardware can speed sense the console device. The console should then print: D1 SLEUTH 3000 (HP D411A.uu.ff) >10 (uu and ff are the current update and fix levels.) You can now enter the dialog necessary to format a disc pack.

The following examples for the various disc subtypes require the DRT number (DRTN) and the unit number (UNIT) to be inserted where applicable. These examples will suffice for the standard operations. However, you should consult your HP Customer Engineer to ensure that there have been no changes to the program before execution of these examples.

Example 1. 2888A Disc Pack

```
>10  DEV 0, DRTN, 14, 100, UNIT
>10  NOPR
>20  RC 0
>30  PR
>40  FMT 0
>50  END
>60  RUN
```

After pack has been formatted, the diagnostic will prompt with:

```
>60
```

Example 2. 7900A Disc Pack

```
>10  DEV 0, DRTN, 13, 100, UNIT
>10  FMT 0
>20  END
>30  RUN
```

After pack has been formatted, the diagnostic will prompt with:

```
>30
```

Example 3. 7905A Disc Pack

```
>10  DEV 0, DRTN, 15, 100, UNIT
>10  FMT 0, 9
>20  END
>30  RUN
```

After the pack has been formatted, the diagnostic will prompt with:

```
>30
```

Example 4. 7920A Disc Pack

```
>10  DEV 0, DRTN, 12, 100, UNIT
>10  FMT 0,9
>20  END
>30  RUN
```

After the pack has been formatted, the diagnostic will prompt with:

```
>30
```

Example 5. 7925A Disc Pack

```
>10  DEV 0, DRTN, 11, 100, UNIT
>10  FMT 0, 9
>20  END
>30  RUN
```

After the pack has been formatted, the diagnostic will prompt with:

```
>30
```

In all of the above examples, when a pack has been formatted, just continue typing in new programs or restart from step 1.

FLEXIBLE DISC PACK FORMATTING ON THE SERIES 30/33/40/44/64

Flexible disc packs can be formatted through the use of the Sleuth Simulator which is a part of the Diagnostic Utility System (DUS). The DUS is on a stand-alone flexible disc and is loaded onto a shutdown system. The Sleuth Simulator is known as SLEUTHSM.

OPERATING INSTRUCTIONS

SYSTEM WITH FLEXIBLE DISC DRIVE

STEP	PROCEDURE
1	Insert the diskette containing the Diagnostic Utility System into the HP 7902 or 9895 flexible disc drive.
2	On the System Control Panel, set the LOAD (Series 30/44), or COLD LOAD (Series 33) thumbwheel switch to the octal value of the DRT number (channel address and device address) of the flexible disc drive.
3	For cold loading from the System Control Panel, press the HALT key and the LOAD (Series (30/44) or COLD LOAD (Series 33) key. For cold loading from the System Console, perform the following steps: (N/A on Series 40/44/64) • While pressing the CNTL key, press the HALT key. • While pressing the CNTL key, press the LOAD key, (Be sure the Front Panel keys are enabled.) At this point DUS is read into memory. This takes approximately 30 seconds.
4	When the cold load is complete, the following message appears on the console: DIAGNOSTIC/UTILITY SYSTEM REVISION xx.xx ENTER YOUR PROGRAM NAME (Type HELP for program information.) :

SYSTEMS WITH MAGNETIC TAPE DRIVE

STEP	PROCEDURE
1	Mount the tape containing the Diagnostic Utility System on the tape drive and place the drive on-line.
2	On the System Control Panel, set the LOAD (Series 30/44), or COLD LOAD (Series 44) thumbwheel switch to the octal value of the DRT number (channel address and device address) of the tape drive.
3	For cold loading from the System Control Panel, press the HALT key and the LOAD (Series 30/44), or COLD LOAD (Series 33) key. For cold loading from the System Console, perform the following steps: (N/A on Series 40/44/64) • While pressing the CNTL key, press the HALT key. • While pressing the CNTL key, press the LOAD key. (Be sure that Front Panel keys are enabled.)
4	At this point the tape containing the DUS is read into memory, and the following message appears on the system console: DIAGNOSTIC/UTILITY SYSTEM REVISION xx.xx ENTER YOUR PROGRAM NAME (Type HELP for program information) :
5	Type "AID" on the console and press RETURN. The following message appears on your console: AID xx.xx >10
6	Type LOAD SLEUTHSM on the console and press RETURN. The Sleuth Simulator is loaded into memory. The System Console displays the next line number in the program and you are ready to enter the dialog necessary to format a disc pack.
7	Enter the following statements to format any disc on the HP 3000 Series 30/33/44 system: >5000 DEV 0,channel#,device#,20,unit# >5010 FMT 0 >5020 RUN

NON-SYSTEM DOMAIN VOLUMES

APPENDIX I

SYSTEM/NON-SYSTEM DOMAINS

All disc drives are divided into two domains at configuration time: system domain and non-system domain. The system domain includes logical device number one and all other disc drives used for such purposes as spooling, etc. These drives will not be used for private volumes. The non-system domain contains all disc drives to be used for private volumes, serial and foreign discs. Disc drives can be changed from the system domain to non-system domain only during a RELOAD. Disc drives can be added to either domain, however, during COOLSTART, COLDSTART, or UPDATE.

ALLOCATING A DISC DRIVE TO THE NON-SYSTEM DOMAIN

Applicable portions of the Initiator-User Dialog, and the appropriate user responses, are listed below to demonstrate how to assign a disc drive to the non-system domain so that it can be used for private volumes. The same procedure would, of course, be used for serial or foreign discs (User response is underlined).

STEP NO.	DIALOG AND USER RESPONSE
4	I/O CONFIGURATION CHANGES? YES
5	LIST I/O DEVICES? YES
8	LOGICAL DEVICE #? Enter the logical device number of the disc drive to be assigned to the non-system domain.
9	DRT #? Enter the DRT entry number of the disc drive to be assigned to the non-system domain.
10	UNIT #? Enter the physical hardware unit number of the device, if the device shares its controller with other devices.
12	TYPE? 0, or 2 for HP 7902 or HP 9895, or 3 for HP 7911, HP 7912, HP 7935, or Integrated Cartridge Tape Unit.
13	SUB-TYPE? Enter 0 for Integrated Cartridge Tape Unit or HP 7902 or HP 9895, 1 for HP 7911, 2 for HP 7912, 4 for HP 7905, 8 for HP 7920 or HP 7935, 9 for HP 7925, 10 for HP 7906.
34	RECORD WIDTH? 128
35	OUTPUT DEVICE? 0
36	ACCEPT JOBS OR SESSIONS? NO
37	ACCEPT DATA? NO
38	INTERACTIVE? NO
39	DUPLICATIVE? NO
41	INITIALLY SPOOLED? NO
43	DRIVER NAME? IOMDISC1, for HP 7905/20/25 on Series II/III. For other drivers on other systems, consult Appendix A.
48	DEVICE CLASSES? PVDISC Any arbitrary classname can be entered.
49	IS PVDISC A SERIAL DISC CLASS? NO This question is asked if this is the first device in PVDISC and it is a supported serial disc device. Enter NO for private volume (non-system) domain. (Classname (if not a system domain disc drive) can be used for private volumes and serial disc if the answer is NO.) A YES answer signifies that the drive will be used for serial disc exclusively.
72	DISC VOLUME CHANGES? YES
83 (q)	NON-SYSTEM VOLUME ON DEVICE # ADD TO SYSTEM VOLUME SET? NO

PRIVATE VOLUMES

Private volumes is a facility that permits removable disc packs, when mounted on disc drives, to be accessed by MPE.

Some features of private volumes are:

- Allows users to access removable disc packs.
- When a private disc volume is mounted on a drive, MPE automatically recognizes the name of the volume and what drive it is on.
- Private volume disc packs can be formatted on line.
- Disc packs can be condensed on line (reclaims free space on a disc volume by making occupied portions of the disc contiguous, thus condensing free space which had existed between occupied portions).
- Private volume disc packs are transportable between HP 3000 Computer Systems.
- Disc-to-disc copy can be performed with private volumes.

VOLUME SET

A volume set is a set of not more than eight removable disc volumes which share a common file directory. Volume sets are assigned and allocated at the group level within an account. A group can be assigned to only one volume set, known as its home volume set.

Every volume set has as one of its members a volume designated as the *master volume* of the set. This volume contains a directory of the volume set. The directory contains such information as a list of all accounts and groups sharing the volume set, all files within the volume set, and a table of volumes for the volume set. The files of a volume set are unique to that volume set, and cannot be shared among other volume sets. For example, if a volume set is defined with the name VOLSET1, then one of the members of this volume set must be the master volume and must be named VOLSET1.

Volume sets can be divided into subdivisions, called *volume classes.* One of the volumes of a volume class must be the master volume of the set, thus a single volume can be a volume class only if it is the master volume. A volume class is identified by a unique name and is accessed as a unit. It is the smallest volume unit that can be referenced by a user. When a user references a volume class, therefore, he is notifying MPE that only a portion of a defined volume set need be mounted to satisfy the anticipated file-access requests. If a volume class consists of more than one volume, all volumes in the class must be mounted in order to access the volume class.

A *home volume set* is the volume set or class assigned to a group when the group is created with the :NEWGROUP command or altered with the :ALTGROUP command. Files belonging to the group are located on the home volume set. A user's home volume set need not be mounted for the user to log on; it need not be mounted until such time as the user attempts to access file space in the set. A user can explicitly request that the volume set be mounted before any file access is attempted, and can release the set after he is through with it. Implicit requests to mount a group's home volume set are invoked any time the user attempts to access file space on that volume set. An implicit request causes the set to be mounted only for the duration of the operation which generated the implicit request.

USING PRIVATE VOLUMES

The following steps must be performed to allow users to access private volumes:

1. One or more disc drives must be allocated to the non-system domain during a RELOAD.
2. A volume set definition must be created with the :NEWVSET command.
3. Each member of the volume set must be conditioned with the Vinit subsystem.
4. The user must be given use private volumes capability (UV) with the :NEWUSER or :ALTUSER command.

CREATING A VOLUME SET/CLASS DEFINITION

Volume sets and classes are defined with the :NEWVSET command by System Managers, or by Account Managers who have the Create Volumes (CV) capability.

The format of the :NEWVSET command is

:NEWVSET *vsname*
;MEMBERS=*vname:type*[,*vname:type*[,. . .[,*vname:type*]]]
[;CLASS=*vcname:vname*[,*vname*[,. . .[,*vname*]]]]

To create a volume set definition named PRIVOL1, log on as MANAGER.SYS (or as Account Manager if you have CV capability), and enter:

:NEWVSET PRIVOL1;MEMBERS=PRIVOL1:HP 7920,SLAVE1:HP 7920,SLAVE2:
HP 7920

This defines a private volume set named PRIVOL1, with a master volume PRIVOL1 and two slave volumes SLAVE1 and SLAVE2. Note that one member, and one member only, in the MEMBERS list must have the name of the volume set. It is this member that is (by definition) designated to be the master volume of the set. The name can appear anywhere in the MEMBERS list.

The foregoing :NEWVSET command creates an entry in the system directory for volume set

PRIVOL1.PUB.SYS

Note that the group (PUB) and account (SYS) are merely part of the identifying name for the volume set (signifying the log-on group and account under which the definition was created) and do not imply that files on this volume set must belong to the PUB.SYS group and account.

If volume classes are to be defined, you could enter:

:NEWVSET PRIVOL1;MEMBERS=PRIVOL1:HP 7920,SLAVE1:HP7905,&
: SLAVE2:HP7920;CLASS=PV1CL1:PRIVOL1,SLAVE2

One of the members of a volume class must be the master volume of the volume set, thus note that in the class PV1CL1 above, PRIVOL1 is a member.

To modify an existing volume set definition, use the :ALTVSET command. The format of the :ALTVSET command is

:ALTVSET *vsname*
[;ADDCLASS=*vcname:vname*[,*vname*[,. . .[,*vname*]]]]
[;EXPANDCLASS=*vcname:vname*[,*vname*[,. . .]]]
[;EXPANDSET=*vname:type*[,*vname:type*[,. . .[,*vname:type*]]]]

To modify volume set definition PRIVOL1, you could enter:

```
:ALTVSET PRIVOL1;EXPANDSET=SLAVE3:HP7920,SLAVE4:HP7920;&
: ADDCLASS=PV1CL2:PRIVOL1,SLAVE3,SLAVE4
```

The above command added two members to volume set PRIVOL1, then created a new volume class definition, PV1CL2, containing the master volume and the two new members SLAVE3 and SLAVE4.

To purge a volume set or class, use the :PURGEVSET command. The format is

:PURGEVSET *vcsname*

where vcsname is volume set/class name. For example, to purge volume set PRIVOL1, you would enter:

```
:PURGEVSET PRIVOL1
```

To purge volume class PV1CL2, you would enter:

```
:PURGEVSET PV1CL2
```

Figure I-1 shows system and non-system domain disc drives, and the relationship of volume members, classes, and sets.

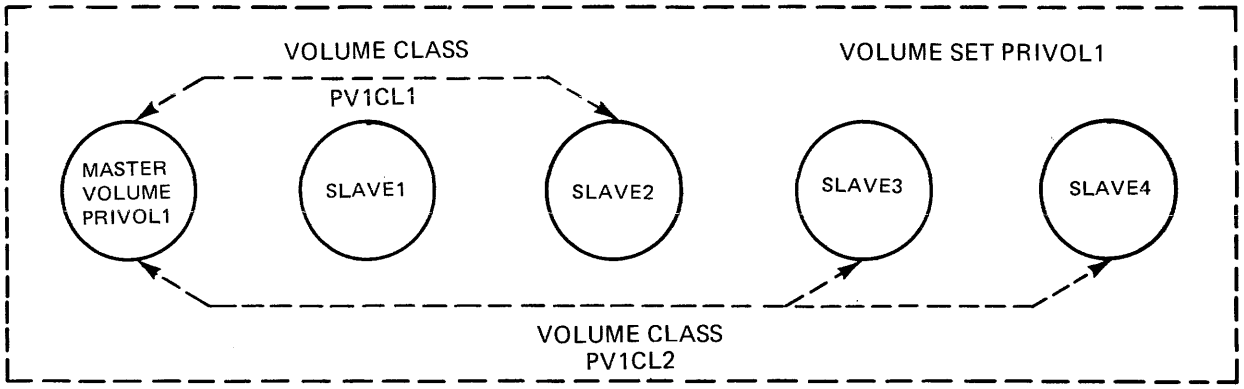
PRIVATE VOLUME CONDITIONING

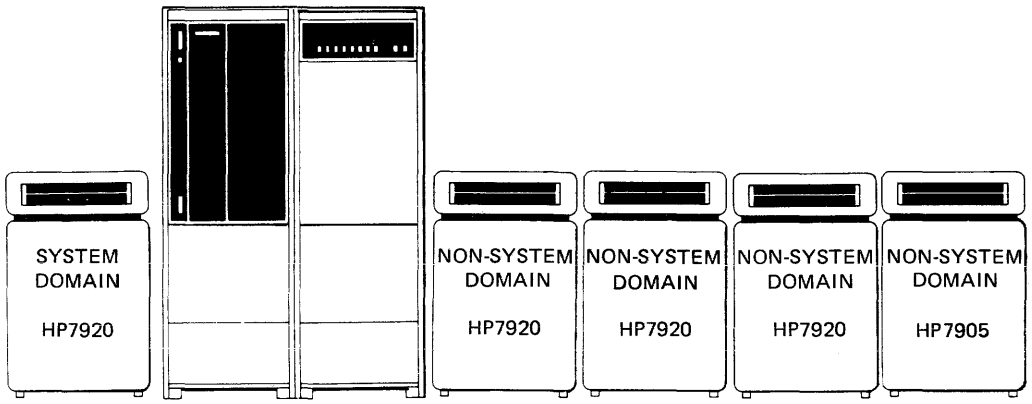
Conditioning of private volumes must be performed by the Vinit subsystem. Unlike the SLEUTH program, which is run offline only, Vinit is run online while the system is up and running.

Vinit can be used to condition private volumes formatted previously with the SLEUTH program, or it can be used to format and condition private volumes not formatted previously by SLEUTH.

ASSIGNING CREATE VOLUMES AND USE VOLUMES CABILITIES TO ACCOUNTS

The use volumes (UV) capability is assigned at the account and user level, thus giving users in that account the capability to access private volumes. The create volumes (CV) capability gives the Account Manager the capability to create private volume set and class definitions.





NOTES

1. Private volumes are removable disc packs.
2. Private volume set resides on non-system domain disc drives.
3. Volume set consists of master volume and up to seven slave volumes.
4. Volume class consists of master volume and may contain slave volumes.
5. Volume class is smallest volume unit that can be referenced by user.

Figure I-1. Private Volume Set

To create a new account named PVUSER, you could enter:

```
:NEWACCT PVUSER,MGR;CAP=AM,AL,GL,SF,ND,UV,CV,IA,BA;&
:VS=PRIVOL1.PUB.SYS:SPAN
```

The above :NEWACCT command creates a new account named PVUSER, with one user named MGR, and a home group named PUB. Included in the capability list are the UV (use volumes) and CV (create volumes) capabilities.

The SPAN parameter of the command enters this account in the system directory and creates an account entry in the volume set's directory. (This is called spanning; that is, an entry is created in both the system directory and the volume set's directory).

The master volume of a volume set (or class) must be mounted when spanning an account into a volume set directory.

To modify an existing account's capabilities to add CV and UV capabilities, you could enter:

```
:ALTACCT OLDACCT;CAP=AM,AL,GL,SF,ND,UV,CV,IA,BA;&
: VS=PRIVOL1.PUB.SYS:SPAN
```

The above :ALTACCT command added the CV and UV capabilities to the account named OLDACCT and created an account entry in PRIVOL1 volume set's directory. (The account still exists in the system directory, the SPAN parameter has no effect on this directory.) Note that the only time you would use the SPAN parameter of the :ALTACCT command is when you plan to assign one or more of the account's groups to a private volume set, or when you plan to create a new group in the account and assign it to a private volume set. An account may contain any mix of system domain groups and private volume groups. The only time a group's files are assigned to a private volume is when the group itself is assigned to that private volume.

ASSIGNING A GROUP TO A PRIVATE VOLUME SET

Use the :NEWGROUP command (must be the Account Manager) to create a new group and assign it to a private volume set. For example,

```
:NEWGROUP PVGROUP1;CAP=IA,BA;VS=PRIVOL1.PUB.SYS:SPAN
```

The above command marks the group entry for group PVGROUP1 in the system directory as a private volume group and creates a group entry for the group in PRIVOL1 volume set's directory.

The master volume of a volume set (or class) must be mounted when spanning a group into a volume set directory.

To create new users for PVGROUP1 and assign UV capability, you could enter:

```
:NEWUSER JOHN;HOME=PVGROUP1;CAP=SF,ND,UV,IA,BA
:NEWUSER JANE;HOME=PVGROUP1;CAP=SF,ND,UV,IA,BA
```

Thus, a new account, PVUSER, and a new group, PVGROUP1, have been created and spanned into the private volume directory located on the master volume for volume set PRIVOL1. Figure I-2 illustrates what happens when an account and group are spanned from the system directory into a private volume set directory.

To modify an existing group to assign the group to a private volume set, use the :ALTGROUP command. For example,

```
:ALTGROUP OLDGROUP;CAP=IA,BA;VS=PRIVOL1.PUB.SYS:SPAN
```

The above command modifies existing group OLDGROUP to mark the group's entry in the system directory as a private volume group, and creates a group entry in PRIVOL1 volume set's directory.

NOTE

No files are transferred to the private volume set. An existing group which contains files cannot be spanned to a private volume set. See below (HOW TO MOVE AN EXISTING GROUP CONTAINING FILES TO A PRIVATE VOLUME) for a description of assigning a group containing files to a private set.

Before an existing group can be assigned to a private volume set with the :ALTGROUP command, the :ALTACCT command must be entered with the SPAN parameter to create an entry for the account in the volume set's directory.

All temporary files will be built on the home volume set. This requires that the volume set be physically mounted. Also, certain MPE commands (such as :SHOWJOB, :STORE, and :RESTORE) build temporary files and can only be used if the volume set is loaded.

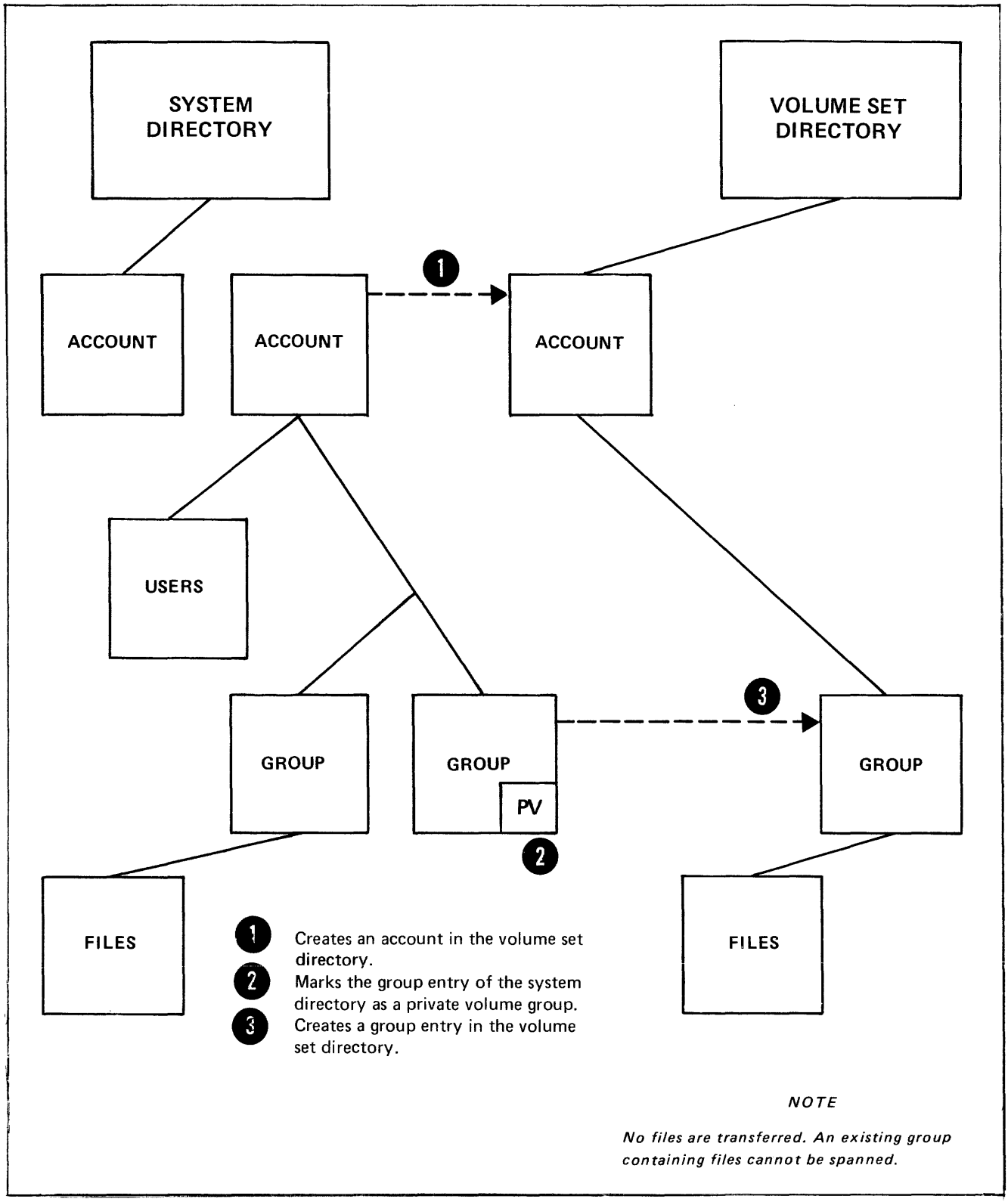


Figure I-2. Spanning

HOW TO MOVE AN EXISTING GROUP CONTAINING FILES TO A PRIVATE VOLUME SET

To assign an existing group and its files to a private volume set, proceed as follows:

1. Store the group's files on tape or serial disc.
2. Purge the files from the group.
3. Enter an :ALTACCT command with the SPAN parameter to create an entry in the private volume set's directory for the account to which the group belongs.
4. Enter an :ALTGROUP command with the SPAN parameter to create an entry for the group in the private volume set's directory.
5. Restore the group's files.

For example,

```
:LISTF @.OLDGRP
FILENAME
PROLL      GFILE    PGUIDE

:FILE STOR;DEV=TAPE
:STORE @.OLDGRP;*STOR
FILES STORED = 3
FILES NOT STORED = 0

  :PURGE PROLL
  :PURGE GFILE
  :PURGE PGUIDE
  :ALTACCT OLDACCT;CAP=AM,AL,GL,SF,ND,UV,CV,IA,BA;&
  : VS=PRIVOL1.PUB.SYS:SPAN
  :ALTGROUP OLDGRP;CAP=IA,BA;VS=PRIVOL1.PUB.SYS:SPAN
  :RESTORE *STOR;@.OLDGRP
FILES RESTORED = 3
FILES NOT RESTORED = 0
```

SUMMARY

In summary, the complete procedure for establishing private volumes and providing users with the capability to access them is as follows:

1. Allocate one or more disc drives to the non-system domain during a RELOAD.
2. Create a volume set definition with the :NEWVSET command.
3. Down a non-system domain disc drive with the console =DOWN command.
4. Mount a disc volume on this downed device.
5. Run the Vinit subsystem.
6. Set the volume to scratch (Vinit SCRATCH command).
7. Format the volume (Vinit FORMAT command).
8. Initialize the volume (Vinit INIT command).
9. Give users use volumes (UV) capability with the :NEWACCT and :NEWUSER commands or :ALTACCT and :ALTUSER commands.

An example of the complete procedure is as follows (assume that the disc drives have already been assigned to the non-system domain):

```
:NEWVSET PRIVOL1;MEMBERS=PRIVOL1:HP7920,SLAVE1:HP7920,&
: SLAVE2:HP7920;CLASS=PV1CL1:PRIVOL1,SLAVE2
=DOWN 3
13:29/8/DISMOUNT ON LDEV# 3
13:33/8/UNKNOWN VOLUME ON LDEV# 3
:VINIT
VINIT SUBSYSTEM (A. 0)
>DSTAT ALL
  LDEV-TYPE     STATUS      VOLUME (VOLUME SET-GEN)
  1-7920        SYSTEM      MH7920U0
  2-7905 (R)    PV-AVAIL    USERVOL1    (USERVOL1.PUB.SYS-0)
  3-7920 (R)    DOWNED      (*.0)
  4-7920 (R)    PV-AVAIL    SLAVOL1     (USERVOL1.PUB.SYS-0)
  5-7920 (R)    PV-AVAIL    SLAVOL2     (USERVOL1.PUB.SYS-0)
  6-7920 (R)    PV-AVAIL    SLAVOL3     (USERVOL1.PUB.SYS-0)
>SCRATCH 3
>FORMAT 3
>INIT PRIVOL1,3,PRIVOL1.PUB.SYS
 ENTER DIRECTORY SIZE (SECTORS) : 500
```

```
>DSTAT ALL
  LDEV-TYPE      STATUS      VOLUME (VOLUME SET-GEN)
  1-7920         SYSTEM      MH7920U0
  2-7905 (R)     PV-AVAIL    USERVOL1    (USERVOL1.PUB.SYS-0)
  3-7920 (R)     DOWNED      PRIVOL1     (PRIVOL1.PUB.SYS-0)
  4-7920 (R)     PV-AVAIL    SLAVOL1     (USERVOL1.PUB.SYS-0)
  5-7920 (R)     PV-AVAIL    SLAVOL2     (USERVOL1.PUB.SYS-0)
  6-7920 (R)     PV-AVAIL    SLAVOL3     (USERVOL1.PUB.SYS-0)
>EXIT
END OF SUBSYSTEM
  :NEWACCT PVUSER,MGR;CAP=AM,AL,GL,SF,ND,UV,CV,IA,BA;&
  : VS=PRIVOL1.PUB.SYS:SPAN
  :HELLO MGR.PVUSER
  :NEWGROUP PVGROUP 1;CAP=IA,BA;VS=PRIVOL1.PUB.SYS:SPAN
  :NEWUSER JOHN;HOME=PVGROUP1;CAP=SF,ND,UV,IA,BA
  :NEWUSER JANE;HOME=PVGROUP1;CAP=SF,ND,UV,IA,BA
```

POINTS TO REMEMBER.

1. A private volume set consists of from one to eight removable volumes.
2. One of the members of a private volume set must be the master volume and must have the same name as the volume set.
3. One of the members of a volume class must be the master volume.
4. The volume class is the smallest volume unit that can be referenced by a user.
5. The master volume of a volume set (or class) must be mounted in order to access the volume set.
6. All members of a volume set (or class) must be mounted in order to span an account or group into the volume set's directory.
7. No files are transferred when an existing group is assigned to a private volume set. A group's files must be stored offline, purged from the group, then restored after the group has been assigned to a private volume set.
8. The following conditions must exist before a volume set can be accessed by a group:
   A. A definition must exist for the volume set.
   B. The group must be spanned into the volume set's directory.
   C. All members of the volume set (or class) must be mounted.
   D. The system directory must be linked to the volume set directory. This is called binding. Binding links the group entry in the system directory with the group entry in the volume set directory so that files belonging to that group can be accessed, and new files created by that group will be assigned to the volume set.

It is possible for a group's home volume set to be mounted but not bound for that group. For example, group A may have volume set A mounted and bound. Group B, however, even though volume set A is also its home volume set, cannot access volume set A until it mounts and binds the volume to its group. This is accomplished with an explicit or implicit mount request. An explicit mount request is generated with the :MOUNT command. The :MOUNT command causes the volume set to be mounted (possibly only logically, it may already be physically mounted) and bound for the group that issued the command. The command remains in effect until it is explicitly released with a :DISMOUNT command. Implicit requests cause the volume set to be mounted and bound only for the duration of the operation which generated the request. Implicit mounts are generated by the following operations:

Running a program
Opening a file
:STORE/:RESTORE commands
:EDITOR command
:BUILD command
:PURGE command
:SAVE command

HOW TO MOVE A VOLUME SET FROM ONE SYSTEM TO ANOTHER

One of the advantages of private volumes is that they are transportable between HP 3000 computer systems. This is accomplished as follows:

1. Create the same volume set definition on all HP 3000 systems which are to share the private volume set.
2. Establish an account with the same name on all HP 3000 systems which are to share the private volume set.
3. On one of the HP 3000 systems, span the account and create and span all groups to the private volume set so that entries for the account and all its groups are entered in the volume set's directory.
4. Create the same groups on the other HP 3000 systems and use the VS = parameter with the :NEWGROUP command to specify the volume set definition. (The groups should have the same home volume set on each system; however, the groups and accounts need be spanned only once to create the entries in the volume set's directory.)

The master volume of a volume set (or class) must be mounted when an account or group is being spanned.

Once the above four steps are completed, the volume set is recognizable to each of the HP 3000 systems and can be transported between any of the systems.

:MOUNT COMMAND. The :MOUNT command requests that your volume set be mounted. The format is

:MOUNT [* / *vcsname*] [*.groupname*[*.acctname*]]

[;GEN=[*genindex*]]

If you are logged on in the same group as the private volume group, the groupname.acctname parameter of the :MOUNT command can be omitted. If you are logged on in a group other than the private volume group, you must specify that group in the mount request so that the correct volume set will be mounted and bound.

If the :MOUNT command is entered in one of the forms

:MOUNT **.*group.account*

Group and account to be mounted (if not logged on in this group)

:MOUNT * (If logged on in group to be mounted)

:MOUNT (If logged on in group to be mounted)

the volume set is bound (the system directory group is linked to the volume set directory group) so that files of that group from the volume set can be accessed.

As an example, if you want to bind group PUB.TEST, which resides on PRIVOL1.PUB.SYS, you would enter:

:MOUNT *.PUB.TEST

If you are logged on under group PUB, you could enter:

:MOUNT*

or

:MOUNT

If the actual volume set name is specified in the :MOUNT command, as in

:MOUNT PRIVOL1.PUB.SYS

the operator will be requested to mount the volume set but no groups will be bound and therefore you cannot access files on the volume set. This form of the :MOUNT command is used before spanning or :STORE and :RESTORE in order to make the set available.

The genindex parameter specifies which generation of the volume set is to be mounted. If omitted or if it is -1, MPE ignores the generation when attempting to satisfy the mount request.

The asterisk (*) specifies the home volume set for the group and account specified, or the log on group and account if groupname.accountname are not specified.

:DISMOUNT COMMAND. The :DISMOUNT command requests the Console Operator to dismount a previously-mounted volume set. The format is

```
:DISMOUNT [ * | vcsname ] [ .groupname[.acctname] ]
```

If you are logged on in the group to be dismounted, you can enter the :DISMOUNT command in one of the forms

```
:DISMOUNT
:DISMOUNT *
```

Otherwise, the group and account must be specified.

The groupname.acctname parameter, if included, specifies the group and account under which the volume set was created.

:LISTVS COMMAND. The :LISTVS command produces a formatted listing of volume set definition information. Three different formats are available, as follows:

```
:LISTVS
 ACCOUNT= SYS        GROUP= PUB

 VOLSET

                class

UCLASS2 (C)    USERVOL1      USERVOL2
```

```
:LISTVS  ,1
ACCOUNT= SYS                GROUP= PUB

VOLSET          MEMBERS         TYPE

UCLASS2 (C)     USERVOL2        HP 7905
USERVOL1        USERVOL1        HP 7905
USERVOL2        USERVOL2        HP 7905
                SLAVOL2         HP 7905
```

```
:LISTVS  ,2
ACCOUNT= SYS                GROUP= PUB

VOLSET          MEMBERS         TYPE        LDEV        STATUS

UCLASS2 (C)     USERVOL2        HP 7905
USERVOL1        USERVOL1        HP 7905
USERVOL2        USERVOL2        HP 7905      2          IN-USE
                SLAVOL2         HP 7905      3          IN-USE
```

:VSUSER COMMAND. The :VSUSER command prints a listing of all users of a currently mounted volume set.

For example,

```
:VSUSER
VOLUME SET NAME          JOBNUM     JOB NAME
USERVOL2.PUB.SYS         #S32       TEST.MPE
```

If no volume sets are in use, :VSUSER returns the message

NO VOLUME SETS IN USE

:DSTAT COMMAND. The :DSTAT command displays the current status of the disc drives on the system. If the ldn parameter is entered with the command, only the disc drive with that ldn is displayed. If ALL is included, all disc drives are displayed. The default if no parameters are included is that only non-system discs are displayed. For example,

```
:DSTAT
 LDEV-TYPE      STATUS      VOLUME (VOLUME SET-GEN)
 2-7905 (R)     MOUNTED     USERVOL2    (USERVOL2.PUB.SYS-0)
 3-7905 (R)     MOUNTED     SLAVOL2     (USERVOL2.PUB.SYS-0)
```

CONSOLE OPERATOR COMMANDS FOR PRIVATE VOLUMES

:VMOUNT COMMAND. The :VMOUNT command enables or disables the Private Volumes Facility. The format of the :VMOUNT command is

$$:\text{VMOUNT} \left\{ \begin{matrix} \text{ON[,AUTO]} \\ \text{OFF} \end{matrix} \right\} \text{[;ALL]}$$

:VMOUNT OFF	Sets the Private Volumes Facility off. All requests, explicit and implicit, are rejected. Requests, even though they cannot be satisfied, are still printed on the console.
:VMOUNT ON	Sets the Private Volumes Facility on. All valid requests are allowed and the operator must reply to such requests.
:VMOUNT ON, AUTO	Sets the Private Volumes Facility on. All valid requests are allowed and MPE attempts to satisfy such requests without operator intervention (no message is sent to the console). If the specified volume set (or class) is physically mounted, MPE logically mounts it. If the specified volume set (or class) is not physically mounted, the mount request is denied.

:VMOUNT nnn ;ALL — Causes all private volume related messages to be printed on the console, including those not requiring operator intervention.

:VSUSER COMMAND. This command lists users of private volume sets. The format of the listing is the same as the :VSUSER command.

:LMOUNT COMMAND. The :LMOUNT command logically mounts a private volume set or class. No binding occurs with this command. The format of the :LMOUNT command is

:LMOUNT vcsname.groupname.acctname[;GEN=genindex]

The groupname.acctname parameter specifies the group and account under which the volume set definition was created. The genindex parameter specifies which generation index of the volume set is to be mounted. If omitted, MPE does not check the generation index of the specified volume set.

:DSTAT COMMAND. The :DSTAT command displays the current status of the disc drives on the system.

EXAMPLES OF USER MOUNTS AND DISMOUNTS AND CONSOLE OPERATOR RESPONSES

If the Console Operator enters the command :VMOUNT ON;ALL, all private volume generated messages are displayed on the console. In addition, the operator must respond to all mount requests.

```
:VMOUNT ON;ALL

:DSTAT
LDEV-TYPE       STATUS      VOLUME (VOLUME SET-GEN)
2-7905 (R)      PV-AVAIL    USERVOL2    (USERVOL2.PUB.SYS-0)
3-7905 (R)      PV-AVAIL    SLAVOL2     (USERVOL2.PUB.SYS-0)
```

If the user enters a :MOUNT command, a message is displayed on the console and the operator is required to answer YES or NO.

```
:MOUNT
?15:33/#S76/19/ACCESS TO USERVOL2.PUB.SYS BY MGR.MPETEST  (Y/N)
 :REPLY 19,Y
:DSTAT
  LDEV-TYPE        STATUS      VOLUME (VOLUME SET-GEN)
  2-7905 (R)       MOUNTED     USERVOL2   (USERVOL2.PUB.SYS-0)
  3-7905 (R)       MOUNTED     SLAVOL2    (USERVOL2.PUB.SYS-0)
```

When a :DISMOUNT command is entered, no intervention is required of the operator but the following message is displayed on the console.

```
:DISMOUNT
15:33/#S76/19/AVAILABLE DRIVE ON LDEV# 2, 3
```

An implicit mount request resulting from the Editor ADD command requires operator intervention (because the Editor builds a work space file in the user's file domain).

```
:EDITOR
HP32201A.7. 0H EDIT/3000 FRI, FEB 10, 1978, 3:33 PM
(C) HEWLETT-PACKARD CO. 1976
/ADD
?15:33/#S76/23/ACCESS TO USERVOL2.PUB.SYS BY MGR.MPETEST    (Y/N)
 :REPLY 23,Y
    1    //
. . .
/END
15:34/#S76/23/AVAILABLE DRIVE ON LDEV# 2, 3

END OF SUBSYSTEM
```

When the Editor is exited, MPE indicates that the drives are available.

The :BUILD command generates an implicit mount request requiring operator intervention. The volumes are mounted only for the duration of the :BUILD command, then MPE displays the device available message.

```
:BUILD FILE1
?15:34/#S76/19/ACCESS TO USERVOL2.PUB.SYS BY MGR.MPETEST  (Y/N)
 :REPLY 19,Y
15:35/#S76/19/AVAILABLE DRIVE ON LDEV# 2, 3
```

Calling the FOPEN intrinsic generates an implicit mount request. This is demonstrated in the following example.

```
:RUN FCOPY.PUB.SYS
15:35/#S76/23/ACCESS TO USERVOL2.PUB.SYS BY MGR.MPETEST    (Y/N)
:REPLY 23,Y
        .
        .
        .
END OF PROGRAM
15:35/#S76/23/AVAILABLE DRIVE ON LDEV# 2, 3
```

Note that again, the mount only lasts for the duration of the operation which generated the implicit mount request and the drive available message is displayed when the dismount occurs.

If a user does not want to be delayed while the operator responds to every mount request, he should enter the :MOUNT command at the beginning of the session and leave the volume set mounted until no longer needed.

For Example,

```
:MOUNT
:DSTAT
  LDEV-TYPE        STATUS      VOLUME (VOLUME SET-GEN)

  2-7905 (R)       MOUNTED     USERVOL2   (USERVOL2.PUB.SYS-0)
  3-7905 (R)       MOUNTED     SLAVOL2    (USERVOL2.PUB.SYS-0)
:EDITOR
HP32201A.7.0H EDIT/3000 MON, FEB 13, 1978, 4:50 PM
(C)  HEWLETT-PACKARD CO. 1976
/ADD
     1
     2//
. . .
/E
END OF SUBSYSTEM

:BUILD FILE1
:RUN FCOPY.PUB.SYS
        .
        .
        .
END OF PROGRAM

:DISMOUNT
15:54/#S34/16/AVAILABLE DRIVE ON LDEV# 2, 3
```

Note that operator intervention was not necessary when the :EDITOR, :BUILD, and :RUN commands were entered.

If the :VMOUNT ON,AUTO command is entered, MPE attempts to satisfy all explicit and implicit mount requests without intervention from the operator. The only time a message is output to the console is when a mount request cannot be satisfied by MPE and operator assistance becomes necessary. Some examples.

```
:VMOUNT ON,AUTO
:DSTAT
  LDEV-TYPE        STATUS      VOLUME (VOLUME SET-GEN)

  2-7905 (R)       PV-AVAIL    USERVOL2   (USERVOL2.PUB.SYS-0)
  3-7905 (R)       PV-AVAIL    SLAVOL2    (USERVOL2.PUB.SYS-0)
:MOUNT
:DSTAT
  LDEV-TYPE        STATUS      VOLUME (VOLUME SET-GEN)

  2-7905 (R)       MOUNTED     USERVOL2   (USERVOL2.PUB.SYS-0)
  3-7905 (R)       MOUNTED     SLAVOL2    (USERVOL2.PUB.SYS-0)
```

The only time operator assistance would be needed is if a physical mount were necessary.

If the :VMOUNT OFF command is entered, the Private Volumes Facility is disabled and mounts are not allowed. For example,

```
:VMOUNT OFF
:MOUNT
PRIVATE VOLUMES FACILITY NOT ENABLED (PVERR 20)
VOLUME SET MOUNT NOT PERFORMED
```

PRIVATE VOLUMES APPLICATIONS

Private volumes, with the capability of mass storage of data and fast recovery of such data, has many applications. Some typical applications are presented in the following paragraphs.

Consider an environment in which a computer system is being used by two or more different types of users. For example, one group of users may consist of members of an engineering department who are using the HP 3000 for calculations and to write and execute programs which demonstrate the feasibility of certain functions. This group may consist of several interactive users, each with moderately large data bases which are used in mathematical model testing.

A separate group with a different application on the same system could be an accounting department or a manufacturing department, each with the large data bases necessary for payroll, order processing, inventory control, etc.

It can be seen that the most efficient use of the system resources will not occur if both the engineering and manufacturing (or accounting) departments are contending for system resources at the same time. The use of private volumes can alleviate this potential problem.

1. Configure the system so that a suitable number of disc drives are allocated to the non-system domain.
2. Establish two accounts, one for engineering and one for manufacturing.
3. Define two private volume sets and assign the groups in each account to a separate volume set.
4. Assign a different time period for each group of users to access the computer system.

Thus, each group of users would load their private volume disc packs on the system at the beginning of their time period, access their files for the duration of this period, and unload the disc packs at the end of the time period.

A second application could be where a program periodically (say once a week or once a month) updates a large data base. If such a data base is left on the system, otherwise useable disc space is not available. Rather than store the files offline on magnetic tape, the files and the program could reside on a private volume set. Then, when the files are to be updated, the private volume set can be mounted and the program run.

SERIAL DISCS

Serial discs are non-system domain discs treated as serial devices. To MPE the discs appear to be magnetic tape drives and thus provide fast backup and recovery capability when used as an alternative to magnetic tape in SYSDUMP, STORE and RESTORE activities.

Any moving-head disc supported by private volumes can be designated for serial use either during a SYSDUMP or during an initialization (cold load) dialog. First, the disc is assigned to a CLASS by logical device number and then the CLASS is designated a serial disc class in response to a special prompt.

NOTE

Only the upper (removable) platter of a HP 7905 or 7906 disc drive may be used as a serial disc or private volume. The lower (fixed) platter must be either configured in the system domain or not configured.

Also, the Integrated Cartridge Tape Unit is treated only as a serial disc device.

Media mounted on a serial disc device must be initialized for serial use by the Console Operator using the VINIT subsystem. Initialization places a special mark in the disc label which is subsequently used for identification by MPE. When MPE recognizes a medium with this mark, it sends a mount notification message to the system console.

It is possible to configure a disc drive into two classes, one of which is a serial disc class and the other which is private volume class. For example,

CLASS NAME	LOGICAL DEVICE NUMBER
PVDISC	5, 6, 7, 8
SDISC	7, 8

All of the above devices (5, 6, 7, 8) can be used for private volumes or serial disc (provided classes PVDISC and SDISC are in the non-system domain). The device class SDISC (assuming that it has been configured as a serial disc class) can be used in, for example, the :FILE and :BUILD commands and either device 7 or 8 will then be allocated. If it is desired to use device 5 or 6 for a serial disc operation, however, the logical device number must be specified.

NOTE

The above is not applicable to the Integrated Cartridge Tape Unit.

CONDITIONING SERIAL DISC MEDIA

Before a medium can be used as a serial disc, it must be conditioned with the VINIT subsystem. The procedure is as follows:

1. Down the drive on which the medium will be conditioned (console :DOWN command).
2. Set the RUN-STOP on the disc drive to STOP. (Not applicable to the Integrated Cartridge Tape Unit.)
3. Set the FORMAT switch (behind the front door of the disc drive) to ON. (Not applicable to the Integrated Cartridge Tape Unit.)
4. Mount the volume on this disc drive, or insert cartridge into drive and wait until it is loaded.
5. Log on (must have SM or OP capability).
6. Run the VINIT subsystem (:VINIT command).
7. Scratch the medium (VINIT "SCRATCH" command).
8. Format the medium (VINIT "FORMAT" command).
9. Condition the volume as a serial disc (VINIT "SERIAL" command).
10. Up the drive before accessing medium (console :UP command).

FOREIGN DISCS

The foreign disc is a type of non-system domain disc, similar to a serial disc and private volume, that enables non-privileged users to access and alter data residing on any removable disc (including the flexible disc) that is not in standard HP 3000 MPE format. For example, you will be able to read or write HP 250, HP 300 or even IBM 3741 formatted diskettes on the Series 30/33/40/44/64 provided that you know the disc and diskette formats. The data is written to or read from a disc volume as though it were a standard MPE file. All that you need to do is write your own programs to interpret (if reading) or format (when writing) the data. (Foreign Disc operations are not applicable to the Integrated Cartridge Tape Unit.)

A query in SYSDUMP and INITIAL will allow the user to assign the device class "foreign". Once a device class has been designated as a foreign class, it is then possible to mount any drive-compatible disc on the corresponding logical device and, by utilizing your own code, you can access and alter any and all information on that disc. Whenever you want to convert any disc format (i.e., serial disc format) to a foreign disc, the volume initialization subsystem, VINIT, is used.

Creating a foreign disc on logical device 3, you would mount a diskette on the drive and enter the following:

```
>FOREIGN 3
```

Logical device 3 must be a non-system domain disc drive in the DOWN state. The action of this command is to fill the MPE label of a disc that was either a previously formatted volume or not, with zeros, since all zeros is not a recognizable MPE label. (Any other nonrecognizable sector zero content would do as well). This volume will then be recognized as a foreign disc with all subsequent usage. (The MPE label area is track 0, sector 0.)

The status of the foreign disc on logical device 3 can be displayed using the VINIT "DSTAT" command:

An example of displaying the status of the foreign disc mounted on logical device 3 (still in the DOWN state) would be:

```
>DSTAT 3
  LDEV-TYPE          STATUS          VOLUME (VOLUME SET-GEN)
  3-7902             DOWNED          *FOREIGN DISC*
```

There are no new user commands. However, the user command "DSTAT" will also display the "FOREIGN" status of a foreign disc, and has the same format as the VINIT "DSTAT" command. A new console operator command, :FOREIGN has been designed to enable the operator to manually intervene when necessary. See the *Console Operator's Manual.*

VINIT SUBSYSTEM

The VINIT subsystem is accessed with the :VINIT command. The format is

VINIT [listfile]

If listfile is specified, output generated by certain VINIT functions is sent to the device specified.

Once accessed, VINIT displays a "greater than" (>) prompt and awaits a command from you. VINIT commands are described in the following paragraphs.

USING THE VINIT SUBSYSTEM

To condition (format and initialize) a private volume:

1. Down the disc drive on which the volume will be conditioned (console=DOWN command).
2. Set the RUN-STOP switch on the disc drive to STOP.
3. Set the FORMAT switch (behind the front door of the disc drive) to ON.
4. Mount the volume on this disc drive.
5. Set the RUN-STOP switch on the disc drive to RUN.
6. Log on (must have SM or OP capability).
7. Run the VINIT subsystem (:VINIT command).
8. Set the volume to scratch (VINIT "SCRATCH" command).
9. Format the volume (VINIT "FORMAT" command).
10. Initialize the volume (VINIT "INIT" command).

INIT COMMAND

The INIT command is used to complete the conditioning of a private volume formatted previously by the SLEUTH program or by the VINIT "FORMAT" command. The INIT command may not be used to condition a disc volume in the system domain. The format is

>INIT *vname,ldn*[,*vsname.groupname.acctname*][;GEN=*gen index*]

vname	Specifies the name of the volume set member (volume) to be conditioned.
ldn	Specifies the logical device number on which the volume is mounted. This parameter must specify a non-system disc drive that is in the DOWN state or SCRATCH.
vsname.groupname.acctname	Specifies the name of the volume set. This parameter (although shown as optional above) must be specified if the volume set name has not been specified in a previous >INIT command.
genindex	A value from 0 to 32767 specifying the generation index of the volume. If not specified, VINIT assigns an index of zero. All volumes of a volume set should be initialized with the same generation index to allow subsequent mountings to be successful.

If vname specifies the master volume of a volume set, additional initializing is performed. Specifically, a volume table of volume members, and a volume set file directory nucleus are placed on the volume.

Before a volume can be initialized, the following conditions must be met:

1. The volume set must have been defined previously.
2. The volume must be physically mounted on the device specified by ldn.
3. The logical device must be a non-system disc drive with a SCRATCH volume mounted, or it must be in the DOWN state. If it is DOWN but not SCRATCH, the user will be prompted to verify that he wishes to destroy the old contents of the pack. (This requirement for DOWN or SCRATCH is called "DOWN-SCRATCH".)
4. The volume must have been formatted previously with the SLEUTH program or with the VINIT "FORMAT" command.

As part of the initializing process, VINIT will construct a Defective Tracks Table. (For the 7912 and 7933 discs, VINIT will construct a Defective Sectors Table which will record suspect sectors.) If defective tracks are encountered during initialization, a message indicating the number of suspect tracks detected is printed. You then should reassign the defective tracks.

This command is inappropriate for the Foreign Disc Facility because foreign discs have no label and therefore no volume name. The *vname* parameter is required for the INIT command. This command is also inappropriate for serial discs.

FORMAT COMMAND

The FORMAT command allows on-line formatting of all private volumes. Formatting a medium is necessary only when a new, previously unused medium is to be initialized or when an irrecoverable pack error has been detected on a previously used medium. The FORMAT command performs these tasks which the SLEUTH program does not:

1. Writes a valid SCRATCH label (see SCRATCH function).
2. Sets the tracks above the logical pack size to the SPARE state.
3. Builds a Defective Tracks Table (DTT) containing tracks which failed the verification part of the formatting process. (For the HP 7911, HP 7912, Integrated Cartridge Tape Unit and HP 7933 discs, builds a Defective Sectors Table (DSCT) containing suspect sectors.) The user is prompted to decide whether to reassign or recover each suspect track. The user may also request that certain tracks be reassigned even though they passed the verification phase if he knows they are defective. HP disc packs come with a list of tracks which have failed a long and extensive diagnostic test. It is conceivable that such tracks may pass the single test of the FORMAT command, but THESE TRACKS SHOULD BE REASSIGNED.

The format of the FORMAT command is

>FORMAT *ldn*

ldn Specifies the logical device number of the device on which the medium is mounted. This parameter must specify a non-system drive which is in the DOWN-SCRATCH state.

NOTE

The FORMAT switch on the disc drive must be ON for the >FORMAT command to be effective. This switch is located behind the front door of the disc drive. See the appropriate Maintenance Manual for the device if you have any questions.

This command is appropriate for Foreign discs.

SERIAL COMMAND

The SERIAL command is used with all serial media. It changes the volume label to indicate that the volume is a serial medium. The format is

>SERIAL *ldn*

ldn Specifies the logical device number of the device on which the medium is mounted. This parameter must specify a non-system serial medium which is in the DOWN-SCRATCH state.

FOREIGN COMMAND

The FOREIGN command allows you to create a foreign disc by filling the label area of a previously formatted disc volume with zeros.

>FOREIGN *ldn*

ldn Specifies the device on which the volume is mounted. The device must be a non-system domain disc drive in the DOWN state. When the disc is mounted the System will automatically recognize it as a foreign disc.

This command is not appropriate for the Integrated Cartridge Tape Unit.

SCRATCH COMMAND

The SCRATCH command allows you to set a removable medium to the scratch state, thus making it available for assignment to a volume set. In addition, the RESET parameter of this command can be used to set a medium from scratch to non-scratch. The format is

>SCRATCH *ldn* [;RESET]

ldn Specifies the logical device number of the device on which the medium is mounted. This parameter must specify a non-system serial medium which is in the DOWN-SCRATCH state.

:RESET Resets a medium from scratch to non-scratch.

This command is inappropriate for foreign discs.

COPY COMMAND

The COPY command copies the contents of one volume to a second volume. This command operates on a volume-by-volume basis. The master volume of the set need not be mounted; however, it is advisable to copy all members of a volume set at the same time. The format is

>COPY *fromldn,toldn*[;GEN=[*genindex*]]

fromldn The logical device number of the device containing the volume to be copied from.

toldn The logical device number of the device containing the volume to be copied to. This parameter must specify a device which is in the DOWN state and which has a volume mounted on it. (This volume must be set to scratch.)

;GEN=*genindex* A value from 0 to 32767 specifying the generation index of the volume. If not specified, the generation index of the to-volume will be one greater than that of the from-volume. If ;GEN= is specified but genindex is not, the generation index of the to-volume will be the same as that of the from-volume. You may specify a generation index for the to-volume that is different than that of the from-volume.

The following conditions must exist for the COPY command to be successful:

1. The to-device must be in the DOWN state or have a volume mounted on it.
2. Both devices must be of the same type (i.e., HP 7920A, etc.)
3. The to-device must have no deleted tracks where file extents exist on the from-device.

The COPY command will place an exact image of the from-volume on the to-volume. No compaction or other re-organization of the file space on the volume will be performed. Areas in the from-pack's Free Space Table are skipped to speed up the copying process.

COPY operates on only one from-volume at a time. Thus, in order to back up a multi-volume set, multiple COPY operations have to be performed. Please note that because the volume set directory is located only on the master volume of a set, it would be of little practical use to back up only one member of a multi-volume set.

Note that the to-volume will not be recognized automatically after copying, thus it must be switched off-line then on-line in order for it to be used.

The COPY command is not applicable to serial discs.

DSTAT COMMAND

The DSTAT command displays the status of one or more serial media. The format is

```
>DSTAT [ ldn
         ALL or @ ]
```

ldn Specifies the logical device number of a particular serial medium.

ALL or @ Status of all serial media on the system is displayed.

Default: If neither *ldn* or ALL is specified, status of all non-system serial media is displayed.

An example of the DSTAT command:

```
>DSTAT ALL
```

LDEV-TYPE	STATUS	VOLUME (VOLUME SET-GEN)	
1-7920	SYSTEM	MH 7920U0	
2-7905 (R)	PV-AVAIL	USERVOL1	(USERVOL1.PUB.SYS-0)
3-7920 (R)	DOWNED	PRIVOL1	(PRIVOL1.PUB.SYS-0)
4-7920 (R)	PV-AVAIL	SLAVOL1	(USERVOL1.PUB.SYS-0)
5-7920 (R)	PV-AVAIL	SLAVOL2	(USERVOL1.PUB.SYS-0)
6-7920 (R)	PV-AVAIL	SLAVOL3	(USERVOL1.PUB.SYS-0)
7-7902	DOWNED	*FOREIGN DISC*	

(The notation (R) after the serial medium type indicates removable (non-system domain).

```
>DSTAT 7
```

LDEV-TYPE	STATUS	VOLUME (VOLUME SET-GEN)
7-7902	FOREIGN	*UNALLOCATED*

This is the message generated when ldn 7 is up.

PDEFN COMMAND

The PDEFN command prints a listing from the system directory of the volume set definition for the specified volume set. The listing contains the following information:

Volume set definition.

Number of volumes in the volume set.

The index numbers, names, and device sub-types of all the member volumes.

The format of the PDEFN command is

```
>PDEFN [ *
         vsname    .group.account]
```

* Specifies the home volume set for the group specified.

vsname Specifies a volume set name.

If neither * nor *vsname* is specified, the listing will be for the volume set specified in the last INIT command.

An example of the PDEFN command:

```
>PDEFN USERVOL2.PUB.SYS
 SET DEFINITION:  USERVOL2.PUB.SYS   MVTAB INDEX:  0
 NUMBER OF VOLUMES:  2  VOLUME MASK:  %000003
 INDEX        MEMBER      SUBTYPE
   1          USERVOL2        4
   2          SLAVOL2         4
```

This command is not applicable to foreign discs.

PLABEL COMMAND

The PLABEL command lists the contents of the label of the medium mounted on the specified logical device. If the medium is a scratch medium, a message signifying this is printed. Otherwise, the following information is printed:

Medium name.

Device type and sub-type.

Creation date and generation index.

Volume set to which the volume belongs. (Private volumes only.)

Master volume set information. (Private volumes only.)

The names of the member volumes of the set. (Private volumes only.)

The format of the PLABEL command is

>PLABEL *ldn*

ldn Specifies the logical device number of the device on which the medium is mounted.

An example of the PLABEL command:

```
>PLABEL2
 LDEV 2
 VOLUME NAME:  USERVOL2, TYPE: 0, SUBTYPE: 4
 CREATE DATE:  1/20/78, GENERATION:  0
 VS NAME: USERVOL2, GROUP: PUB , ACCOUNT: SYS
 MASTER VOLUME INFORMATION —
   DIR. BASE:  48 , DIR.SIZE:  500
 VOLUME DIRECTORY
   NAME       SUBTYPE

 USERVOL2        4
 SLAVOL2         4
```

This command is inappropriate for Foreign Disc Facility. The information generated is incorrect and is not taken from the label. Instead, the following message is generated:

"Inappropriate operation for Foreign Disc."

PFSPACE COMMAND

The PFSPACE command will print a listing of the Disc Free Space Map in one of two formats: either as a histogram of free space areas or as a report of the free space area addresses and sizes.

The format of the PFSPACE command is

```
>PFSPACE { ldn [;ADDR] }
         { ALL        }
```

ldn Specifies the logical device on which the volume is physically mounted.

;ADDR Displays the Disc Free Space entries, in free space address/free space size format, for *ldn*.

ALL Displays, in histogram format, the Disc Free Space Map for all system and private volumes which are physically mounted.

PFSPACE *ldn* will print out a histogram of the free space on *ldn*. If ADDR is specified, it will print out a map of the free space addresses and sizes. PFSPACE ALL will print out a histogram of all the system and private volume discs which are physically mounted on the system.

PFSPACE is inappropriate for foreign discs and serial discs.

An example of PFSPACE output:

```
>PFSPACE 2
VOLUME DFST                    LDEV 2
LARGEST FREE AREA= 194444
  SIZE      COUNT    SPACE     AVERAGE
>100000     1        194444    194444
>10000      0        0         0
>1000       0        0         0
>100        0        0         0
>10         0        0         0
>1          100      200       2
TOTAL FREE SPACE= 194644
```

An example of PFSPACE output when the ADDR keyword is used:

```
>PFSPACE 1;ADDR
LDEV: 1
 ADDRESS    SIZE     ADDRESS    SIZE     ADDRESS    SIZE
   36736       1       38454       1       43056       1
   44620       1       70307       1       74696     158
   75355  120245
 NO. ENTRIES:  7
 TOTAL VOLUME CAPACITY:  195600 SECTORS
 TOTAL FREE SPACE AVAILABLE:  120408 SPACE
 MAXIMUM CONTIGUOUS AREA:  120245 SECTORS
```

PDTRACK COMMAND

The PDTRACK command allows you to print a list of all suspected, reassigned and deleted tracks. The format is:

>PDTRACK *ldn*

ldn Specifies the logical device number of the device on which the volume is mounted.

This command is inappropriate for foreign discs and serial discs.

COND COMMAND

The COND command allows you to compact the specified disc by making the occupied portions of the disc contiguous. The format of the VINIT COND command is

>COND *ldn* [;RECOVER]

ldn Specifies the logical device number of the device on which the volume to be condensed is physically mounted.

;RECOVER For private volumes ONLY. This is equivalent to the "RECOVER LOST DISC SPACE" operation of INITIAL.

A COND operation can be performed on either a system volume or a private volume. Serial and foreign discs cannot be CONDensed.

Once a COND operation has begun, the VINIT subsystem cannot be aborted until the operation is complete. If COND is performed on a private volume, the volume set for that private volume must be mounted.

COND will condense all disc space, with the exception of areas which contain open permanent disc files, temporary files (including $OLDPASS), and deleted tracks. If any temporary files are open when a system failure occurs, those areas will not be recovered by COND. To reclaim these areas, use the "RECOVER LOST DISC SPACE" option of INITIAL for system volumes, or "COND *ldn* ;RECOVER" for private volumes.

If the system crashes during a COND operation, you do not have to re-initialize the disc and restore the files (or in the case of a system disc, you do not have to RELOAD): the files will be intact. There may be a loss of disc space, but this can be recovered with the "RECOVER LOST DISC SPACE" option of INITIAL for system discs, of "COND *ldn* ;RECOVER" for private volumes.

When the RECOVER keyword is used with COND, the user of VINIT must have the only session of job running on the system and have no temporary files. If any open permanent files are encountered while RECOVERing with COND, COND will abort. The specified *ldn* should be re-initialized and the files should be restored to insure the integrity of the disc.

COND *ldn* ;RECOVER has two phases. In the first phase, all space that has been allocated to permanent files is reclaimed. The message, "BEGINNING RECOVER," will appear to indicate the beginning of this phase. When all the space for permanent files has been reclaimed, the first phase is complete and the message "FINISHED RECOVERING FILES, BEGINNING COND" appears. The second phase is the COND operation. If the system crashes before the later message appears, allocation (the creation of new files) will be disabled on this *ldn* when the system is brought up. The COND *ldn* ;RECOVER operation can then be restarted. If the system crashes after the later message appears, there will be a loss of disc space, but the private volume will not have to be re-initialized.

If COND *ldn* ;RECOVER encounters any fatal errors, allocation is disabled on *ldn*. This is to prevent access to that disc. If this occurs, the disc should be re-initialized and the files should be restored to insure the integrity of the disc.

If COND *ldn* ;RECOVER encounters a file which has space that has already been allocated to another file, the file will be purged from the directory.

The COND function checks that there are no suspect tracks on any member of the volume set (system volume set or private volume set). Any such tracks must be disposed of by DTRACK for private volumes or INITIAL for system volumes.

COND will disable system logging. If system logging is enabled before the COND command is issued, a message will be printed on the system console. When the operation completes, system logging will be re-enabled.

The COND command is inappropriate for foreign discs and serial discs.

DTRACK COMMAND

The DTRACK command allows you to process defective tracks and defective sectors. The format is

>DTRACK *ldn*

ldn Specifies the logical device number of the device on which the volume is mounted.

An error encountered during normal access to the volume will cause the accessing program to abort. Later, you can use the VINIT DTRACK command to assign an alternate for the suspect track or mark it defective.

A suspect track causes the defective tracks table for the volume to be examined. For each suspect track, you will be prompted with either

WARNING — IN DIRECTORY
or
WARNING — IN FREE SPACE MAP

followed by

SUSPECT TRACK LDN #n TRACK %m. RECOVER OR REASSIGN?

If your response is RECOVER, the entry is removed from the DTT and the next entry is processed; you have chosen to "ignore" the error. If your response is REASSIGN, VINIT attempts to save the data in the suspect track.

If a defective track occurs in the Disc Free Space Map area, COND;RECOVER is done to rebuild the map, but no files are purged.

If a defective track occurs in the directory, you will be warned and the volume set will be scratched.

Should COND ;RECOVER be required, you will need to mount the entire volume set.

The HP 7933 disc utilizes a Defective Sectors Table (DSCT) rather than a Defective Tracks Table. When a suspect sector is enountered, it is recorded in the DSCT and the transfer is terminated.

DTRACK tries to recover all these bad sectors by sparing with retained data. After the sparing is done the entry is removed from the DSCT.

(The algorithm for sparing is based on the inclusion of a spare sector on each track of the discs. This spare sector is used for the first sparing operation on that track. Subsequent sparing operations performed for sectors on that track cause the entire track to be spared. It is not known until after a sparing operation is performed whether the entire track was spared.)

If data was lost during the spare operation, the action taken depends on what kind of data occupied the area:

If data from a file is lost permanently, you will get a listing of the file's name and the physical and logical addresses of all of the bad sectors in that file. You will then have the option to purge all the files with lost data or to go through a dialog which will ask if each individual file with lost data should be saved.

If the data lost was part of the Disc Free Space Map, you should perform the COND; RECOVER operation to rebuild the DFSM.

If the data lost was part of the volume set directory, the volume set should be scratched.

This command is inappropriate for foreign discs and serial discs.

EXIT COMMAND

The EXIT command exits the VINIT subsystem. The format is

>EXIT

HELP COMMAND

The HELP command prints a listing of the VINIT commands displaying the command name, syntax, and a brief definition of that command. The format is:

```
>HELP
```

An example of the HELP command:

```
:VINIT
VINIT B00.02 (C) HELETT-PACKARD CO., 1978
>HELP
 FORMAT ldev<<writes a SCRATCH label; sets up FST, DFSM, DTT, DSCT, spares>>
 SERIAL ldev <<writes a serial disc type label>>
 INIT volumename, ldev, vsid.gname.aname[;GEN=xx] <<writes PV label>>
 INIT volumename,ldev[;GEN=xx] <<uses last VS from INIT or PDEFN>
 SCRATCH ldev; [RESET]  <<writes(resets) a SCRATCH type label>>
 COPY from-ldev,to-ldev[;GEN=xx] <<copies volumes of the same type>>
 COND ldev;RECOVER  <<condenses disc space holes and recovers lost disc space for
                                        PV>>
 DSTAT [ALL:@:ldev]  <<gives status of drives>>
 PDEFN vsname.gname.acctname <<lists VS specified>>
 PDEFN *.gname.acctname <<lists home VS of gname.acctname>>
 PDEFN  <<lists home VS of logon group (or repeats lasts VS)>>
 PLABEL ldev <<prints disc label>>
 PDTRACK ldev  <<prints defective track table contents or defective sector table
                                        contents>>
 PFSPACE ldev  <<prints disc free space map>>
 DTRACK ldev  <<processes suspect tracks in defective track table>>
<EXIT

END OF SUBSYSTEM
:
```

The following example demonstrates using the VINIT subsystem to format and initialize the master volume of the new volume set PRIVOL1. Note that because the master volume of the set is being initialized, VINIT asks for the directory size (in sectors). Enter a value between 384 and 65,000. The value you enter here will depend on the number of members in the volume set. The discussion of system directory sizes in Appendix E can be used as a guide. The value of 500 sectors was used in the example and should suffice for most volume sets.

```
:DOWN 3
13:29/8/DISMOUNT ON LDEV# 3
13:33/8/UNKNOWN VOLUME ON LDEV# 3

:VINIT
VINIT SUBSYSTEM (A. 0)
>DSTAT ALL
  LDEV-TYPE      STATUS    (VOLUME SET-GEN)

  1-7920         SYSTEM    MH7920U0
  2-7905 (R)     PV-AVAIL  USERVOL1    (USERVOL1.PUB.SYS-0)
  3-7920 (R)     DOWNED    (*.0)
  4-7920 (R)     PV-AVAIL  SLAVOL1     (USERVOL1.PUB.SYS-0)
  5-7920 (R)     PV-AVAIL  SLAVOL2     (USERVOL1.PUB.SYS-0)
  6-7920 (R)     PV-AVAIL  SLAVOL3     (USERVOL1.PUB.SYS-0)
>SCRATCH 3
>FORMAT 3
>INIT PRIVOL1, 3,PRIVOL1.PUB.SYS
  ENTER DIRECTORY SIZE (SECTORS) : 500
>DSTAT ALL
  LDEV-TYPE      STATUS    VOLUME (VOLUME SET-GEN)

  1-7920         SYSTEM    MH7920U0
  2-7905 (R)     PV-AVAIL  USERVOL1    (USERVOL1.PUB.SYS-0)
  3-7920 (R)     DOWNED    PRIVOL1     (PRIVOL1.PUB.SYS-0)
  4-7920 (R)     PV-AVAIL  SLAVOL1     (USERVOL1.PUB.SYS-0)
  5-7920 (R)     PV-AVAIL  SLAVOL2     (USERVOL1.PUB.SYS-0)
  6-7920 (R)     PV-AVAIL  SLAVOL3     (USERVOL1.PUB.SYS-0)
>EXIT

END OF SUBSYSTEM
```

Table I-1. Messages Generated by VINIT

BEGINNING VERIFY.
SPARING SECTOR ! (DECIMAL)
CARTRIDGE HAS RUN OUT OF SPARE SECTORS. (PVERR 191)
MEDIA IS UNINITIALIZED. (PVERR 192)
MEDIA HAS NOT BEEN 'FORMAT'ED. USE FORMAT COMMAND AND THEN USE 'SERIAL' AGAIN. (PVERR 193)
YOU HAVE DONE A 'SCRATCH;RESET' OPERATION ON A CARTRIDGE THAT WAS PREVIOUSLY IN 'SCRATCHED' STATE OR OTHERWISE UNRECOGNIZABLE TO MPE. THIS OPERATION MAY NOT BE MEANINGFUL. (PVWARN 194)
CARTRIDGE TAPES DO NOT HAVE DEFECTIVE SECTOR TABLES. (PVWARN 195)
UNABLE TO READ THE DISC LABEL. THE FOLLOWING DEFECTIVE SECTOR INFORMATION MAY NOT BE VALID. (PVWARN 196)
UNABLE TO READ THE DEFECTIVE SECTOR TABLE. (PVERR 197)
DEFECTIVE SECTOR TABLE IS EMPTY.
! DEFECTIVE SECTORS FOUND. (SECTOR VALUES ARE IN DECIMAL.)
SECTOR !

INDEX

INDEX (continued)

INDEX (continued)

SERIES 30/33/40/44/64 SYSTEMS

APPENDIX J

There are a number of places in this manual in which variations between Series II/III and Series 30/33/40/44/64 systems are described. Below is a list of pages on which differences are discussed. The users of Series 30/33/40/44/64 systems are encouraged to read these pages to ensure correct usage of the systems.

Part No. 30000-90014
Printed in U.S.A. 12/81
Fifth Edition
E1281

HEWLETT PACKARD