HP 3000 Series II Computer System

System Manager/ System Supervisor

Reference Manual

HEWLETT PACKARD

5303 STEVENS CREEK BLVD., SANTA CLARA, CALIFORNIA 95050

Part No. 30000-90014
Product No. 32002A

Printed in U.S.A. 2/77

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.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

LIST OF EFFECTIVE PAGES

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

Pages	Effective Date
Title	Feb 1977
ii	Jun 1976
iii to iv	Feb 1977
v to x	Jun 1976
1-1 to 1-4	Jun 1976
2-1 to 2-2	Jun 1976
2-3	Sep 1976
2-4	Jun 1976
2-5 to 2-6	Sep 1976
2-7 to 2-11	Jun 1976
2-12	Sep 1976
2-12A to 2-17	Jun 1976
2-18	Jan 1977
2-19	Sep 1976
2-20 to 2-24	Jun 1976
2-25	Sep 1976
2-26 to 2-31	Jun 1976
2-32 to 2-32 A	Sep 1976
2-33 to 2-39	Jun 1976
2-40	Sep 1976
2-41 to 2-42	Jun 1976
2-43	Sep 1976
2-44 to 2-46	Jun 1976
3-1 to 3-3	Jun 1976
3-4	Sep 1976
3-5 to 3-6	Jun 1976
3-7	Sep 1976
3-8 to 3-14	Jun 1976
4-1	Jun 1976
4-2	Sep 1976
4-3	Jun 1976
4-4	Sep 1976
4-5 to 4-8	Jun 1976
5-1 to 5-9	Jun 1976
6-1 to 6-2	Jun 1976
6-3	Sep 1976
6-4 to 6-5	Jun 1976
6-6	Jan 1977
6-7 to 6-12	Jun 1976
6-13	Sep 1976
6-14 to 6-15	Jun 1976
6-16	Sep 1976
6-17 to 6-17 C	Jan 1977
6-17D to 6-20A	Sep 1976
6-21	Jun 1976
6-21A to 6-21B	Sep 1976
6-22 to 6-24	Jun 1976
6-25	Jan 1977
6-26 to 6-36	Jun 1976
6-37	Jan 1977
6-38 to 6-49	Jun 1976
6-50	Jan 1977
6-51 to 6-52A	Sep 1976
6-53	Jun 1976
6-53A to 6-54	Sep 1976
6-55 to 6-63	Jun 1976
6-64 to 6-64B	Jan 1977
6-65	Jun 1976
6-66	Sep 1976
6-67 to 6-69	Jun 1976
6-70 to 6-71	Sep 1976
6-72 to 6-78	Jun 1976
6-79 to 6-81	Sep 1976
6-82 to 6-105	Jun 1976
A-1	Sep 1976
A-2	Jan 1977
B-1 to B-4	Jun 1976
B-5	Sep 1976
B-6	Jun 1976
C-1	Sep 1976
C-2	Jan 1977
C-3	Sep 1976
D-1 to D-11	Jun 1976
E-1	Sep 1976
F-1 to F-5	Jun 1976
G-1 to G-3	Jun 1976
H-1 to H-2	Jun 1976
I-1 to I-4	Sep 1976

PRINTING HISTORY

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

First Edition. . . . . Jun 1976
Updates Nos. 1 and 2 Incorporated. . . . . Feb 1977

PREFACE

This manual, one of the set of manuals that document the Multiprogramming Executive (MPE-II) Operating System, is an "Administrative Level" manual. The Manual Plan on the next page indicates the position of this manual (shaded block) in the overall set.

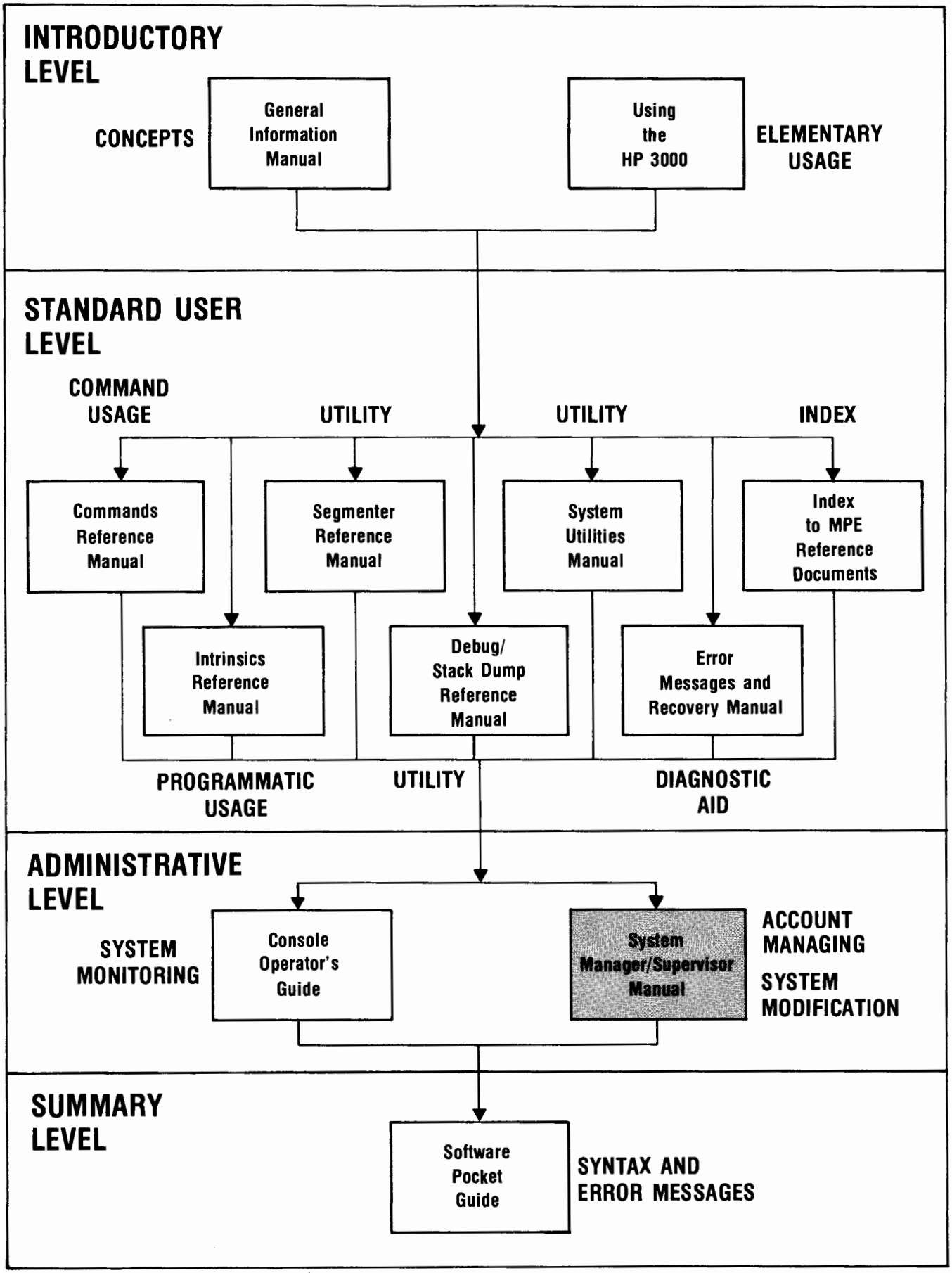
This manual is intended for HP 3000 Series II Computer System users assigned the System Manager, Account Manager, or System Supervisor capabilities.

This manual is a reference book rather than a tutorial text for new programmers. You should understand the basic operating principles of the HP 3000 Series II Computer System. You should also examine the *MPE Operating System, General Information Manual* (part no. 30000-90008) for an overview of the interrelationships between the main hardware and software features offered.

Other manuals which should be available for reference are:

MPE Intrinsics Reference Manual, part no. 30000-90010
MPE Commands Reference Manual, part no. 30000-90009

MANUAL PLAN


INTRODUCTORY LEVEL
CONCEPTS
General Information Manual
Using the HP 3000
ELEMENTARY USAGE
STANDARD USER LEVEL
COMMAND USAGE
UTILITY
UTILITY
INDEX
Commands Reference Manual
Segmenter Reference Manual
System Utilities Manual
Index to MPE Reference Documents
Intrinsics Reference Manual
Debug/ Stack Dump Reference Manual
Error Messages and Recovery Manual
PROGRAMMATIC USAGE
UTILITY
DIAGNOSTIC AID
ADMINISTRATIVE LEVEL
SYSTEM MONITORING
Console Operator's Guide
System Manager/Supervisor Manual
ACCOUNT MANAGING
SYSTEM MODIFICATION
SUMMARY LEVEL
Software Pocket Guide
SYNTAX AND ERROR MESSAGES

CONVENTIONS USED IN THIS MANUAL

NOTATION	DESCRIPTION
[ ]	An element inside brackets is *optional*. Several elements stacked inside a pair of brackets means the user may select any one or none of these elements. Example: $\begin{bmatrix} A \\ B \end{bmatrix}$ user may select A or B or neither
{ }	When several elements are stacked within braces the user **must** select one of these elements. Example: $\left\{ \begin{matrix} A \\ B \\ C \end{matrix} \right\}$ user must select A or B or C.
italics	Lowercase italics denote a parameter which must be replaced by a user-supplied variable. Example: CALL *name* *name* one to 15 alphanumeric characters.
underlining	Dialogue: Where it is necessary to distinguish user input from computer output, the input is underlined. Example: NEW NAME? ALPHA1
superscript C	Control characters are indicated by a superscript C Example: Y^C
return	*return* in italics indicates a carriage return
linefeed	*linefeed* in italics indicates a linefeed
. . .	**A horizontal ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted.**

CONTENTS

ILLUSTRATIONS

TABLES

INTRODUCTION TO COMMANDS

SECTION I

NOTE

This section is an abbreviated version of Section I of the *MPE Commands Reference Manual*. It is assumed that you are already acquainted with the contents of that manual and the manner in which commands are written.

MPE commands allow you to initiate, control, and terminate the processing of programs and to request various other system operations. You generally use them 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 given 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 system onto magnetic tape for backup and modification purposes (:SYSDUMP command).
- Controlling timeslice and priorities for executing processes (:QUANTUM command).

Other commands and functions are available as well; complete specifications for all of them appear 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 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 is now ready for another command.

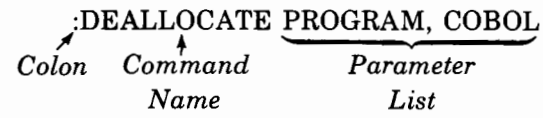
COMMAND ELEMENTS

Table 1-1 gives a summary of command elements and conventions used in this manual.

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 typical command including all three elements appears 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 column 1 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-alphabetic 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. 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 a continuation character (ampersand), the command is continued onto the next record.

NOTE

When you are running programs in batch job mode, bear in mind that MPE scans all 80 columns on 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 interactive mode but entered by you in batch mode).

You can continue commands for up to 255 characters, including prompting colons and continuation ampersands. In continuing a command onto another line, you must not divide a command name, keyword, positional parameter, or keyword sub-parameter; MPE does not permit any such element to span more than one line.

MPE does not begin interpretation of a command until the last record of the command is read. Then, it joins together all records within the command and replaces all prompt characters and continuation ampersands with blanks.

COMMAND ERRORS

If you make an error while entering a command, MPE will inform you of this through a message transmitted to your standard list device. The meaning of such messages and the way in which MPE handles errors are described in the *MPE Error Messages Manual.*

INVOKING COMMANDS PROGRAMMATICALLY

This manual employs two approaches in describing MPE commands and operations: *reference specifications* and *detailed text.*

The *reference specifications*, appearing in Section II, primarily cover the rules for entering each command. Specifically, they show the command syntax and format; define the parameters and discuss constraints upon them and default values for them; and present examples illustrating proper command entries.

The text discussion, appearing in Sections III through VII, explains the functions available through the commands, when and why you would need them, and how to request them through sequences of commands. It also deals with the possible errors that you might encounter and how to avoid them.

Both the reference specifications and the text discussion provide complete cross-referencing, so that you can relate the discussion of an operation or command that appears in one area to the associated material in another.

Table 1-1 presents a summary of command elements and conventions.

Table 1-1. Summary of Command Elements and Conventions

ELEMENTS OF COMMAND FORMAT

Leading Colon:	is the prompt/command identifier character in interactive sessions. is the command identifier character in batch jobs.
Command Name:	is shown in CAPITAL LETTERS IN REGULAR (ROMAN) TYPE, contains no blanks, is delimited by a non-alphabetic character (usually a blank).
Parameters:	are shown in CAPITAL LETTERS IN REGULAR TYPE when they are literal information that you always enter exactly as shown; are 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 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 beginning of list) :COMMANDNAME p1 (from end of list)
Keyword Parameters:	are separated by semicolons and can appear in any order.
Mixed Parameters:	positional parameters are given first; first keyword indicates end of positional list.

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 sessions, jobs, during break, or programmatically).
- Whether interruptable (with BREAK key).
- Capability required (System Supervisor (OP), System Manager (SM), or Account Manager (AM)).
- Examples.
- Where described in text portion of this manual.

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

OP	System Supervisor
SM	System Manager
AM	Account Manager

Table 2-1 lists the commands by functional grouping.

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	:STORE :RESTORE
System Performance Management	:QUANTUM
System Reports and Information Management	:REPORT :RESUMELOG :SHOWLOG :SHOWQ :SWITCHLOG
System Modification and Back-up	:SYSDUMP

:ALLOCATE

Permanently allocates a program file or procedure in virtual memory.

SYNTAX

:ALLOCATE [*code*,] *name*

PARAMETERS

code A directive specifying whether a program file or a code segment containing a procedure is to be allocated.

If PROGRAM is used for *code*, the program file indicated by the *name* parameter is allocated.

If PROCEDURE is used for *code*, the code segment containing the specified procedure indicated by the *name* parameter and residing in SL.PUB.SYS is allocated.

If the *code* parameter is omitted, the default assigned is PROGRAM. (Optional Parameter.)

name The name of the program file or procedure to be allocated. (Required parameter.)

USE

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

NOTES

The user issuing the :ALLOCATE command must pass all security provisions for any file referenced by the *name* parameter of this command.

EXAMPLES

To permanently allocate a program residing on a program file named RUNNER, in the PUB group of the SYS account, you enter:

```
:ALLOCATE RUNNER.PUB.SYS
```

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

```
:ALLOCATE PROCEDURE, PROC1
```

TEXT DISCUSSION

Page 6-82.

: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]]
```

PARAMETERS

acctname — The name of the account to be altered. (Required parameter.)

password — The account password. (This password is used for verifying log-on access only.) If omitted, no password is assigned. (Optional parameter.)

filespace — The disc storage limit, in sectors, for the permanent files of the account. The maximum value permitted is 2 billion sectors. The *filespace* limit cannot be less than the number of sectors currently in use for the account. If omitted, the default is the value assigned when the account was created. (Optional parameter.)

cpu — The limit on total central processor time, in seconds, for the account. (This limit is checked against the actual time accumulated only when a job/session is initiated — thus, the limit never causes a job/session to abort once it is in progress.) The maximum value permitted is approximately $2^{31} - 1$ (2,147,483,647) seconds. If omitted, the default is the value assigned when the account was created. (Optional parameter.)

connect — The limit on total session connect-time, in minutes, allowed the account. Like the *cpu* limit, this limit is checked against actual usage only at log-on time. The maximum value allowed is 2^{31} minutes. If omitted, the default is the value assigned when the account was created. (Optional parameter.)

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

User Attributes:

System Manager	=	SM	
Account Manager	=	AM	(Assigned automatically to the user named during account creation.)
Account Librarian	=	AL	
Group Librarian	=	GL	
Diagnostician	=	DI	
System Supervisor	=	OP	

File-Access Attributes:

Permanent Files	=	SF
Access of non-sharable I/O devices	=	ND

Capability-Class Attributes:

Process-Handling	=	PH	
Extra Data Segments	=	DS	
Multiple RINS	=	MR	
Privileged Mode	=	PM	
Remote Job Entry	=	CS	
Interactive Access	=	IA	At least one of these must be specified.
Local Batch Access	=	BA	

If omitted, the default is the capability set assigned when the account was created. (Optional parameter.)

fileaccess The restrictions on file access pertinent to this account, entered in the same format as the *modelist:userlist* parameter in the :ALTSEC Command described in the *MPE Commands Reference Manual*. If omitted, the default is the file access set assigned when the account was created. (Optional parameter.)

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. If omitted, the default is the subqueue assigned when the account was created. (Optional parameter.)

localattribute The local attribute of the user, as defined at the installation site. This is a double-word 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. If omitted, the default is double-word 0 (null) assigned when the account was created. (Optional parameter.)

USE

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

NOTES

When altering the *capabilitylist* for the SYS account, the SM capability cannot be removed. If *acctname* is SYS, and the *fileaccess* parameter is omitted the following default security is assigned: (R, X: ANY; A, W, L: AC).

:ALTACCT

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. (See table below.) When changing one capability in a *capabilitylist* presently containing several non-default values, you must re-specify the *entire* new *capabilitylist* — not just the changed 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 the 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 the change in advance so that no conflicts will arise.

Parameter	Default Values
password	Null (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, CS, 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	Null (double-word 0)

EXAMPLE

To change an account named AC2 so that its password is GLOBALX and its filespace is limited to 50,000 sectors, a System Manager User could 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]]
```

PARAMETERS

groupname The name of the group whose attributes are to be changed. (Required parameter.)

password The new password to be assigned to the group. (This password is used for verifying log-on access only.) (Optional parameter.)

capabilitylist The list of capabilities, mutually separated by commas, 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	At least one of these
Local Batch Access	=	BA	must be specified.

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 later altered, the group could be left with excessive capabilities. (*System Manager Users should be aware of this when changing an account's capability list.*)

The default assigned is IA, BA. (Optional parameter.)

filespace The 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, nor can it be changed to be less than the number of sectors currently allocated for the group. 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.

cpu The limit on total central processor time, in seconds, for the group. (This limit is checked against the actual time accumulated only when a job/session is initiated — thus, the limit never causes a job/session to abort once it is in progress.) The maximum value permitted is approximately $2^{31} - 1$ (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 later be changed so that some of its groups are left with a limit that exceeds the new account limits. (Optional parameter.)

connect The limit on total session connect-time, in minutes, allowed the group. Like the *cpu* limit, this limit is checked against actual usage only at log-on time. The maximum value allowed is approximately 2 billion 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* limit can later be changed so that some of its groups are left with a limit that exceed the new account limits. (Optional parameter.)

fileaccess The restrictions on file access pertinent to this group, entered in the same format as the *modelist:userlist* parameter in the :ALTSEC Command described in the *MPE Commands Reference Manual*. (Optional parameter.)

USE

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

NOTES

When an entire keyword parameter group 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. (See table below.) In this regard, the values may be the defaults explicitly assigned when the account was created or when the specific group was created.

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 so that no conflicts will arise.

Parameter	Default Values
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.
Note: These defaults are system assigned, provided that account defaults were initially assigned by the system.	

EXAMPLE

To assign a new password of PASS2 to a group named GROUPXX, the Account Manager enters:

```
:ALTGROUP GROUPXX; PASS=PASS2
```

TEXT DISCUSSION

Page 5-2.

:ALTUSER

Alters the 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 (in the :NEWUSER command or the first user assigned during a :NEWACCT command). (Required parameter.)

password — The new password to be assigned to the user. (Optional parameter.)

capabilitylist — The same meaning as the *capabilitylist* parameter for the :NEWACCT command, but applicable only to this user. This parameter cannot specify capabilities not assigned to the account (in :NEWACCT or :ALTACCT commands.) However, if the account is altered 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. (Optional parameter.)

subqueuename — The same meaning as the *subqueuename* parameter for the :NEWACCT command, but applicable only to this user. The priority specified in :NEWUSER cannot be greater than that specified in :NEWACCT or :ALTACCT. (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. 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. (Optional parameter.)

localattribute — The local attribute defined at the installation; this must be a subset of the bit configuration for the *account's* local attribute. (Optional parameter.)

homegroupname — The name of an existing group to be assigned as the homegroup for this user. (Optional parameter.)

USE

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

NOTES

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

When a parameter is modified through :ALTUSER, it immediately takes effect in the directory, but does not apply to users currently logged on; it will take effect for the *next* log-on by those users. For this reason, it is recommended that the users involved be advised of the change in advance so that no conflicts will arise. It is *not* recommended that the *capabilitylist* or *homegroupname* of the MANAGER.SYS user be altered.

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, the user must specify a log-on group when initiating a job or a session.	

EXAMPLE

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

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

The new capabilities for user JONES are a subset of the account capabilities.

TEXT DISCUSSION

Page 5-4.

:DEALLOCATE

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

SYNTAX

:DEALLOCATE [*code,*] *name*

PARAMETERS

code A directive specifying whether a program file or procedure code segment is to be deallocated.

If PROGRAM is used for *code*, the program file indicated by *name* parameter is deallocated.

If PROCEDURE is used for *code*, the code segment containing the specified procedure indicated by *name* parameter residing in SL.PUB.SYS is deallocated.

If *code* is omitted, PROGRAM is assumed. (Optional parameter.)

name The name of the program file or procedure to be deallocated. (Required parameter.)

USE

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

NOTES

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

TEXT DISCUSSION

Page 6-82.

Sets or changes the default execution priority for batch jobs and sets a maximum execution priority for both 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 on his :JOB card. This parameter may be ES, DS, CS, or 0. When the parameter is 0, no limit is imposed on batch jobs. The initial default for the system is DS. (Optional parameter.)

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 card. If *defaultsubqueue* is omitted, there will be no change in the execution default. The initial default for the system is DS. (Optional parameter.)

USE

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

NOTES

The first parameter takes precedence over the second. That is, if the default parameter is greater than the maximum priority parameter, jobs will be initiated with the maximum priority parameter.

Users can still request priority greater than that of the default priority; however, they cannot obtain an execution priority greater than that of the maximum priority setting. If an attempt is made to do so, the maximum priority allowed will be used, and a warning printed to the user indicating that he has requested an execution priority greater than that which is currently allowed by the system.

:JOBPRI

The :JOBPRI command can be a valuable tool in "tuning" the system to meet the requirements of particular load periods on the system. For example, during peak load periods, where there is a heavy interactive load on the system, it is suggested that the default and maximum priorities both be set to DS for batch jobs. This allows the best possible response times to interactive users on the system. When the peak load period is over, the :JOBPRI command can be issued to raise the maximum execution priority so that batch jobs can run in any subqueue the users have requested.

To further increase interactive performance, the System Supervisor should use the IA/BA capabilities (next paragraph) in conjunction with the =LIMIT and :JOBPRI commands. (The =LIMIT command is discussed in the Console Operator's Guide.)

The use of the IA/BA (Interactive Access/Batch Access) allows absolute control over which modes program files run in. It is of particular importance to the System Supervisor because the programs and subsystems under his control generally have the greatest impact on system performance. Thus, for example, if it is desirous to support a large number of terminals with fast interactive response times, the System Supervisor should give all or most of his large subsystems (i.e., SPL, COBOL, RPG, FORTRAN) BA capability only. Of course, strictly interactive programs and subsystems should always be given IA capability.

The following chart is intended to assist System Supervisors in configuring their systems for maximum performance. The chart assumes that all large subsystems have been modified or :PREPed to exhibit BA capability only.

INTERACTIVE USERS	=LIMIT JOBS	:JOBPRI DEFAULT	MAX
0-2	3	CS	0
2-4	2	DS	CS
5-6	2	DS	DS
7-16	1	DS	DS
Over 16	1	DS	DS

EXAMPLES

To raise the maximum execution priority so that batch jobs can run in any subqueue requested, the System Supervisor User enters:

```
:JOBPRI 0
```

TEXT DISCUSSION

none

Lists the attributes for one account, or for all accounts in the system.

SYNTAX

:LISTACCT [@ / *acctname*] [,*listfile*]

PARAMETERS

@ — An indication that the attributes for all accounts are to be listed. Restricted to System Manager Users. (Optional parameter.)

acctname — The name of the account whose attributes are to be listed. *Must* be specified by Account Manager Users who do not *also* have System Manager Capability; furthermore, such users can only specify their own account. (Optional parameter for users with System Manager Capability. Required parameter for users with Account Manager Capability only.)

listfile — The actual designator of the file on which the attribute listing is produced. The options assigned this file when it is opened (through the FOPEN intrinsic) are:

NEW, ASCII, VARIABLE, CCTL, OUTPUT, EXCL

The file is closed (through the FCLOSE intrinsic) with SAVE disposition. If this parameter is omitted, the job/session list device ($STDLIST) is assigned. You may define a disc file with the above attributes for permanent online storage. (Optional parameter.)

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes; aborts
Capability required:		SM or AM

NOTES

If neither @ nor *acctname* is specified by a System Manager User, all accounts (@) are listed.

A sample listing and explanation of the octal decoding is given in Appendix D.

EXAMPLE

To list the attributes of his own account, named ACCTNO23, the Account Manager User enters:

:LISTACCT ACCTNO23

TEXT DISCUSSION

Page 4-4.

:LISTGROUP

Prints a list of attributes for various groups.

SYNTAX

:LISTGROUP [*groupset*] [,*listfile*]

PARAMETERS

groupset The set of groups to be listed, specified as follows:

Specification	Meaning
groupname.acctname	The group named, in the account named.
groupname	The group named, in the log-on account.
@.*acctname*	All groups in the account named.
@	All groups in the log-on account.
@.@	All groups in all accounts. (When this is done, the attributes of the accounts to which the groups belong are also listed.)

If Account Manager Users who are not *also* System Manager Users specify an account, it must be their own.

If *groupset* is omitted, @ is assigned by default.

(Optional parameter.)

listfile The actual designator of the file on which the attribute listing is produced. The options assigned this file when it is opened (through the FOPEN intrinsic) are:

NEW, ASCII, VARIABLE, CCTL, OUTPUT, EXC

The file is closed (through the FCLOSE intrinsic) with SAVE disposition. If the *listfile* parameter is omitted, the job/session list device ($STDLIST) is assigned. You may define a disc file with the above attributes for permanent online storage. (Optional parameter.)

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes; aborts
Capability required:		SM or AM

NOTES

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 PROJECTS, you enter:

```
:LISTGROUP @.PROJECTS
```

TEXT DISCUSSION

Page 4-4.

:LISTUSER

Lists the attributes currently assigned to various users.

SYNTAX

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

PARAMETERS

userset The set of users to be listed, specified as follows:

Specification	Meaning
username.acctname	The user named, in the account named.
username	The user named, in the log-on account.
@.*acctname*	All users in the account named.
@	All users in the log-on account.
@.@	All users in all accounts. (When this is done, the attributes of the user's accounts are also listed.)

If Account Manager Users who are not *also* System Manager Users specify an account, it must be their own.

If *userset* is omitted, @ is assigned by default.

(Optional parameter.)

listfile The actual designator of the file on which the attribute listing is produced. The options assigned this file when it is opened (through the FOPEN intrinsic) are:

NEW, ASCII, VARIABLE, CCTL, OUTPUT, EXCL

The file is closed (through the FCLOSE intrinsic) with SAVE disposition. If the *listfile* parameter is omitted, the job/session list device ($STDLIST) is assigned. You may define a disc file with the above attributes for permanent online storage. (Optional parameter.)

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes; aborts
Capability required:		SM or AM

NOTES

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, the Account Manager User enters:

:LISTUSER CLANCY.MGT

TEXT DISCUSSION

Page 4-4.

:NEWACCT

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

SYNTAX

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

PARAMETERS

acctname The name (maximum of eight alphanumeric characters) assigned to the new account. (Required parameter.)

mgrname The name of the account manager user; he is always the first user created under the account. He receives the following attributes:

User password:	Null (no password).
User capability list:	Same as account's capabilities.
Maximum scheduling priority:	Same as account's maximum priority.
Local attribute:	Same as account's local attribute.
Home group:	PUB

(The attributes of an Account Manager User may be changed later through the :ALTUSER command; in no case, however, is this user granted *effective* attributes greater than those of the account.) (Required parameter.)

password The account password. (This password is used for verifying log-on access only.) If omitted, no password is assigned. (Optional parameter.)

filespace The disc storage limit, in sectors, for the permanent files of the account. The maximum value permitted is 2 billion sectors. If omitted, unlimited permanent file space is assigned. (Optional parameter.)

cpu The limit on total central processor time, in seconds, for the account. (This limit is checked against the actual time accumulated only when a job/session is initiated — thus, the limit never causes a job/session to abort once it is in progress.) The maximum value permitted is approximately 2 billion (2,147,483,647 = $(2^{31} - 1)$) seconds. If omitted, unlimited central processor time is assigned. (Optional parameter.)

connect

The limit on total session connect-time, in minutes, allowed the account. Like the *cpu* limit, this limit is checked against actual usage only at log-on time. The maximum value allowed is approximately 2 billion minutes. If omitted, unlimited connect-time is assigned. (Optional parameter.)

capabilitylist

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

User Attributes:

System Manager	=	SM	
Account Manager	=	AM	(Assigned automatically to the account and manager name in the account.)
Account Librarian	=	AL	
Group Librarian	=	GL	
Diagnostician	=	DI	
System Supervisor	=	OP	

File-Access Attributes:

Permanent Files	=	SF	
Access of non-sharable I/O devices	=	ND	

Capability-Class Attributes:

Process-Handling	=	PH	
Extra Data Segments	=	DS	
Multiple RINS	=	MR	
Privileged Mode	=	PM	
Remote Job Entry	=	CS	
Interactive Access	=	IA	At least one of these must be specified.
Local Batch Access	=	BA	

If the *capabilitylist* parameter is omitted, the following capabilities are assigned by default:

AM,AL,GL,SF,ND,IA,BA

(Optional parameter.)

fileaccess

The restrictions on file access pertinent to this account. The default value for all accounts is (R,A,W,L,X: AC). (Optional parameter.)

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. If none is specified, the CS subqueue is assigned. (Optional parameter.)

localattribute

The local attribute of the user, as defined at the installation site. If omitted, a double-word 0 is assigned. This is a double-word 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 process through the WHO intrinsic for use in the programmer's own security provisions. (Optional parameter.)

:NEWACCT

USE

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

NOTES

In the :NEWACCT command, the default parameters noted are assigned:

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

The Public Group is initially assigned a null password and the same capability-class attributes, permanent file space limit, central processor time limit, and connect 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. (These access provisions are defined syntactically as (R, X: ANY; A, W, L, S: AL, GU). This notation is explained fully in Section III.

EXAMPLE

Suppose that a System Manager wants to create an account with the following characteristics:

Account name = ACI
Manager's name = SMYTHE

(All other parameters are assigned by default.)

To accomplish this, the System Manager User could enter:

:NEWACCT ACI, SMYTHE

The account structure as created would then be as follows:

Account

acctname	ACI
password	Null (no password)
filespace	Unlimited
cpu	Unlimited
connect	Unlimited
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 (no password)
capabilitylist	IA, BA
filespace	Unlimited
cpu	Unlimited
connect	Unlimited
fileaccess	(R, X: ANY; A, W, L, S: AL, GU)

User

mgrname	SMYTHE
password	Null (no password)
homegroup	PUB
subqueuename	CS
localattribute	Null (double-word 0)
capabilitylist	AM, AL, GL, SF, ND, IA, BA

TEXT DISCUSSION

Page 4-4.

:NEWGROUP

Creates a new group within an account.

SYNTAX

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

PARAMETERS

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

password — The group password. (This password is used for verifying log-on access only.) If omitted, no password is assigned. (Optional parameter.)

capabilitylist — A list of *capability-class attributes*, consisting of any or all of the following: IA, BA, PM, MR, DS, or PH (as specified in the discussion of the :NEWACCT command). 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 later altered, the group could be left with excessive capabilities. (System Managers and Account Managers should be aware of this when changing an account's capability list.)

The default assigned is IA, BA. (Optional parameter.)

filespace — The 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. The default value is the account's *filespace*. (Optional parameter.)

cpu — The limit on total central processor time, in seconds, for the group. (This limit is checked against the actual time accumulated only when a job/session is initiated — thus, the limit never causes a job/session to abort once it is in progress.) The maximum value permitted is approximately 2 billion (2,147,483,647 = $(2^{31} - 1)$) 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 later be changed so that some of its groups are left with a limit that exceeds the new account limits. The default value is the account's *cpu* limit. (Optional parameter.)

connect The limit on total session connect-time, in minutes, allowed the group. Like the *cpu* limit, this limit is checked against actual usage only at log-on time. The maximum value allowed is approximately 2 billion 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* limit can later be changed so that some of its groups are left with a limit that exceeds the new account limits. The default value is the account's *connect* limit. (Optional parameter.)

fileaccess The restrictions on file access pertinent to this group. If omitted, the specification assigned by default is:

For Public Group (PUB): (R,X:ANY; A,W,L;S: AL,GU)
For all other groups: (R,A,W,L,X,S: GU)

(Optional parameter.)

USE

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

NOTES

When a keyword parameter or keyword parameter group is omitted from the :NEWGROUP command, the corresponding default parameter takes effect.

EXAMPLE

The Account Manager wants to create a new group named GNAME. The following command is entered:

:NEWGROUP GNAME

Provided that the account was created with the system defaults, the newly created group and account structure would be:

:NEWGROUP

Account

acctname	ACI
password	Null (no password)
filespace	Unlimited
cpu	Unlimited
connect	Unlimited
capabilitylist	AM, AL, GL, SF, ND, IA, BA
fileaccess	(R, A, W, L, X: AC)
subqueuename	CS
localattribute	Null (double-word 0)

Group

groupname	GNAME
password	Null (no password)
capabilitylist	IA, BA
filespace	Unlimited
cpu	Unlimited
connect	Unlimited
fileaccess	(R, A, W, L, X, S: GU)

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 The user password. (This password is used for verifying log-on access only.) If omitted, no password is assigned. (Optional parameter.)

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

User Attributes:

System Manager	=	SM	
Account Manager	=	AM	(Assigned automatically to the account and manager name in the account.)
Account Librarian	=	AL	
Group Librarian	=	GL	
Diagnostician	=	DI	
System Supervisor	=	OP	

File-Access Attributes:

Permanent Files	=	SF	
Access of non-sharable I/O devices	=	ND	

Capability Class Attributes:

Process-Handling	=	PH	
Extra Data Segments	=	DS	
Multiple RINS	=	MR	
Privileged Mode	=	PM	
Remote Job Entry	=	CS	
Interactive Access	=	IA	At least one of these must be specified.
Local Batch Access	=	BA	

This parameter cannot specify capabilities not assigned to the Account (in :NEWACCT or :ALTACCT commands.) However, if the account is

later altered, 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. If the *capabilitylist* parameter is omitted from the :NEWUSER command, the following capabilities are assigned by default: SF,ND,IA,BA. (Optional parameter.)

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. The priority specified in :NEWUSER cannot be greater than that specified in :NEWACCT or :ALTACCT. (But, if the user's account is later altered, 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. 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. The default value is CS. (Optional parameter.)

localattribute The local attribute defined at the installation; this must be a subset of the bit configuration for the *account's* local attribute. If omitted, a double-word 0 is assigned. While it is not involved in standard MPE security provisions, it is available to process through the WHO intrinsic for use in the programmer's own security provisions. (Optional parameter.)

homegroupname The name of an existing group to be assigned as the homegroup for this user. If omitted, no home group is assigned, and so the user must always specify a group when he logs on. (Optional parameter.)

USE

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

NOTES

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 as LHSMITH, and assign him a password of SMITTY and a homegroup of HOMGPX, the Account Manager User issues the following command. (The user will have the default capabilities and "CS" subqueue priority.)

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

Provided that the account and group were created with the system defaults, the newly created user, homegroup, and account structure would be:

Account

acctname	ACI
password	Null (no password)
filespace	Unlimited
cpu	Unlimited
connect	Unlimited
capabilitylist	AM, AL, GL, SF, ND, IA, BA
fileaccess	(R, A, W, L, X: AC)
subqueuename	CS
localattribute	Null (double-word 0)

Group

groupname	Homegrpx
password	Null (no password)
capabilitylist	IA, BA
filespace	Unlimited
cpu	Unlimited
connect	Unlimited
fileaccess	(R, A, W, L, X, S: GU)

User

username	LHSMITH
password	SMITTY
capabilitylist	SF, ND, IA, BA
subqueuename	CS
localattribute	Null (double-word 0)
homegroupname	HOMGPX

TEXT DISCUSSION

Page 5-4.

:PURGEACCT

Removes an account (and its groups (files) and users) from the system.

SYNTAX

:PURGEACCT *acctname*

PARAMETERS

acctname The name of the account to be deleted. (Required parameter.)

USE

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

NOTES

When the :PURGEACCT command is entered in a session, MPE prints a verification request. This aids you in making sure you do not accidentally delete the wrong account. The format of the verification request is

ACCT *accountname* TO BE PURGED?

Respond with YES or NO.

(No verification is made when the :PURGEACCT command is entered in a job.)

If no users are logged-on under the account, and none of its groups/files are in use (as defined in the note below), :PURGEACCT removes the entire account. Otherwise, :PURGEACCT 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. In such cases, to completely purge the account, the user must re-issue the :PURGEACCT command when the account is inactive. It is conceivable that the system would require a restart in order to close files which were left open.

The system account, SYS, cannot be purged. However, if the command is given, 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. For this reason, it is not advisable to attempt this operation, as such files as SPL, FORTRAN, COBOL, etc. could be purged unless currently in use.

NOTE

A file is in use if it is currently opened (FOPEN intrinsic), contains a program being executed (:RUN command or CREATE intrinsic), or is being stored (:STORE command) or restored (:RESTORE command).

A group is in use if any of its files are in use, if it is being used as a home group, or if a :LISTF@ command is applied to its files.

An account is in use if any user is logged on under it; any of its groups are in use; or any of the following commands are applied to it:

:LISTGROUP @
:LISTUSER @
:REPORT @
:RESETACCT

EXAMPLE

When a System Manager User removes an account named ACCTXYZ from the system during a session, the following dialogue takes place:

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

The colon prompt indicates that the account is purged.

TEXT DISCUSSION

Page 4-8.

:PURGEGROUP

Removes a group (and all files belonging to it) from the system.

SYNTAX

:PURGEGROUP *groupname*

PARAMETERS

groupname The name of the group to be removed. (Required parameter.)

USE

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

NOTES

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

GROUP *groupname* TO BE PURGED?

The user verifies his intent to delete the group by responding YES; otherwise, he responds NO.

(No verification is made when the :PURGEGROUP command is entered in a job.)

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, to completely purge the group, the user must re-issue the :PURGE command when neither the group nor its remaining files (if any) are in use. It is conceivable 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. However, if the command is entered, all files which are not system files are purged. For this reason, it is not advisable to attempt this operation. Files such as SPL, FORTRAN, COBOL, etc. could be purged unless currently in use.

When a group is purged, it is recommended that the Account Manager advise in advance all persons with *usernames* that have the group as a *homegroupname*, in order to avoid conflicts.

EXAMPLE

When an Account Manager User removes a group named EXGROUP from the system, this dialogue takes place:

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

The colon prompt indicates that the group is purged.

TEXT DISCUSSION

Page 5-7.

:PURGEUSER

Removes a user from the account.

SYNTAX

```
:PURGEUSER  username
```

PARAMETERS

username The 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

NOTES

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

USER *username* TO BE PURGED?

The Account Manager User must verify his intent by responding YES or NO.

(When the :PURGEUSER command is entered in a job, no verification is made.)

If a :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 user, MANAGER.SYS, cannot be purged.

Removal of a user may leave some files in an account without their creators, preventing anyone from renaming those files or altering their security. But the Account Manager User still can access those files, and can copy or purge them.

EXAMPLE

When an Account Manager User removes the user SMITH from the system, this dialogue takes place:

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

The colon prompt indicates that the user was purged.

TEXT DISCUSSION

Page 5-8.

:QUANTUM

Changes the quantum or priority limits of a circular subqueue.

SYNTAX

:QUANTUM [*quantum*[,[*tpri*][,[*cpri*][,[*dpri*]]]]]

PARAMETERS

quantum The new quantum, in milliseconds, for this subqueue. This value must lie between 0 and 32,767 milliseconds. (Optional parameter.)

tpri The maximum priority limit (lowest priority number) for circular subqueues CS and DS. (Optional parameter.)

cpri The minimum priority limit (highest priority number) for circular subqueue CS. (Optional parameter.)

dpri The minimum priority limit (highest priority number) for circular subqueue DS. (Optional parameter.)

NOTE

At least *one* of the parameters *quantum, tpri, cpri,* and *dpri must* be supplied. The values for *tpri, cpri,* and *dpri* may range from 150 through 250.

USE

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

NOTES

This command is used primarily in on-line tuning of the system to best accommodate the current load.

Misuse of this command can significantly degrade system operating efficiency.

The *quantum* parameter specifies the number of milliseconds a process can retain access to the central processor before its priority is lowered by the system. The amount by which the priority is lowered depends upon the process' priority class. After this adjustment, the process may continue to use the central processor. But if the process looses control of the central processor and must wait to use it again, the system automatically increments its priority. A *quantum* of

about 1/2 second (400 — 600 milliseconds) usually produces efficient system operation. (If *quantum* is too large, system response may become erratic; if it is too small, excessive swapping may result because process-switching occurs too frequently. In either case, system performance is degraded.)

The *tpri* parameter determines the priority assigned to a user process when that process resumes execution after reading terminal input. This parameter should be set at or near *150*. To favor response to system commands, set *tpri* somewhat larger (for instance, at 152 or 153). Significantly larger *tpri* values yield unpredictable results, depending on system demand and configuration.

The *cpri* parameter specifies the lowest priority that the system can assign to a CS process whose time quantum expires. When *cpri* is close to *tpri*, long transactions are favored over short ones in the CS subqueue. When *cpri* is far from *tpri*, short transactions are generally favored over long ones in this subqueue. Setting *cpri* less than *tpri* or greater than *dpri*, however, produces unpredictable results.

The *dpri* parameter indicates the lowest priority that the system can grant a DS process whose time-quantum expires. With *dpri* close to *tpri*, DS processes receive increased throughput, possibly at the expense of response to CS processes. As the distance between *cpri* and *dpri* is increased, the impact of a DS process on CS processes is decreased, CS process response may improve, and DS process throughput may be somewhat reduced.

EXAMPLE

To change the quantum to 500 milliseconds, the System Supervisor user enters:

```
:QUANTUM 500
```

TEXT DISCUSSION

Page 6-81.

:REPORT

Displays accounting information for accounts and groups.

SYNTAX

:REPORT [*groupset*] [,*listfile*]

PARAMETERS

groupset Entries that specify the accounts and groups for which the resource-use report is generated, as described below. If no entry is included, the report covers all groups in the log-on account. (Optional parameter.)

listfile An actual file designator from the output set that denotes the file to which the report is written. If omitted, the job/session listing device (\$STDLIST) is used. For *listfile*, the options supplied (by the FOPEN intrinsic) when the file is opened are:

NEW, ASCII, VARIABLE, CCTL, OUTPUT, EXCL

If the *listfile* parameter is omitted, the job/session list device (\$STDLIST) is assigned. You may define a disc file with the above attributes for permanent online storage. (Optional parameter.)

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes; aborts
Capability required:		SM or AM

NOTES

The *groupset* entries used to specify the accounts and groups covered by the report are denoted below.

Parameter Entry	Group Covered by Report	Account Covered by Report	Who may enter this Combination
groupname	The group named.	The log-on account.	SM or AM
@	All groups.	The log-on account.	SM or AM
groupname.acctname	The group named.	The account named.	SM
@.acctname	All groups.	The account named.	SM
@.@	All groups.	All accounts.	SM
(omitted)	All groups.	The log-on account.	AM
(omitted)	All groups.	All accounts.	SM

:REPORT

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 explained in the text discussion.

In the counts of central processor and connect time used, the values shown reflect time accumulated prior to *the current log-on; the amounts accumulated by the current job/sessions are not included.*

If the type of file for *listfile* is not specified, ASCII is assigned by default.

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

To create permanent *listfiles*, the :FILE command should be used as shown in the examples below.

EXAMPLES

To create a permanent ASCII file on disc (REPFILE) showing resource-usage counts and limits for all groups in the log-on account, the Account Manager User enters:

```
:FILE REPFILE; SAVE; REC=66
:REPORT @, *REPFILE
```

To create a permanent binary file on disc (REPFILE) showing resource usage counts and limits for all accounts/groups in the system, the System Manager User enters:

```
:FILE REPFILE; SAVE; NOCCTL; REC=17,,,BINARY
:REPORT @.@, *REPFILE
```

To show resource usage counts and limits for all accounts/groups in the system and have the listing sent to a line printer, the System Manager User enters:

```
:FILE REPFILE; DEV=LP
:REPORT @.@, *REPFILE
```

TEXT DISCUSSION

Page 4-7.

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

SYNTAX

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

PARAMETERS

@ — Specifies that counters for all accounts/groups are to be reset. (Optional parameter.)

acctname — The name of a particular account whose account and group usage counters are to be reset. (Optional parameter.)

CPU — A specification that *only* the central processor usage counter is to be reset. (Optional parameter.)

CONNECT — A specification that *only* the connect-time usage counter is to be reset. (Optional parameter.)

USE

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

NOTES

If neither @ nor *acctname* is specified, the designated counters for all accounts (and their groups) are reset (@). If neither CPU nor CONNECT is specified, both the central processor and connect-time counters for the designated account (and its groups) are reset. If all parameters are omitted, all counters for all accounts and their groups are reset.

EXAMPLE

A System Manager User can reset the central processor unit resource use counter for all accounts in the system by entering:

```
:RESETACCT @, CPU
```

TEXT DISCUSSION

Page 4-7.

:RESTORE

Reads back into the system, onto disc, any file, or set of files that has been stored off-line (on tape) by :STORE or :SYSDUMP.

SYNTAX

:RESTORE *tapefile* [;[*filesetlist*] [;KEEP] [;DEV=*device*] [;SHOW] [;FILES=*maxfiles*] . . .]

PARAMETERS

tapefile — The name of the tape file on which the *filesetlist* to be retrieved now resides. This file must be referenced in the **formaldesignator* format. A message is output to the console operator telling him which tape to mount, and the device it should be mounted on. (Required parameter.)

filesetlist — The file, fileset, or filesetlist to be restored from tape. Each file is restored as a permanent file. The parameter is written in the same format, and subject to the same constraints as the *filesetlist* parameter of the :STORE command. The number of filesets specified is limited as follows: up to 10 by account name; up to 15 by account name and group name; up to 20 by account name, group name, and file name. If the *filesetlist* parameter is omitted, the default value is @.@.@. (all files on the tape). (Optional parameter.)

KEEP — A specification that if a file referenced in *filesetlist* currently exists on disc, the file on disc is kept and the corresponding tape file is not copied into the system. If KEEP is omitted, and an identically-named file exists in the system, that file is replaced with the one on the tape. If KEEP is omitted, *and* a file on tape is eligible for restoring, *and* a file of the same name exists on disc, *and* this disc file is busy, the disc file is kept and the tape file is not restored. (Optional parameter.)

device — The device class name of the devices or logical device number of the device to which all files are to be restored. The files are marked as having been created on that device class or logical device number. If this parameter is omitted, an attempt is made to restore each file on the same device class (or logical device number) as that on which the file was created. If this attempt fails (e.g., if the device class is non-existent or if the tape was made on previous versions of the operating system) then each file will be replaced onto a disc of the same type (fixed or moving head) and subtype as when created originally. If this fails, then the file will be restored in device class DISC. (Optional parameter.)

SHOW — A request that the names of the restored files be listed. If SHOW is omitted, only the total number of files restored, a list of the files not restored (and the reason they were not restored), and a count of the files not restored, are listed. (Optional parameter.)

maxfiles — The maximum number of files that may be restored. If omitted, the number 4000 is used by default. (Optional parameter.)

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes; aborts
Capability required:		See below

NOTES

A user with System Manager or System Supervisor capability can restore any file from a :STORE tape, assuming the account and group to which the file belongs, and the user who created the file, exist in the system. A user with account-manager capability can restore any file in his account (but cannot restore those with negative file codes unless he also has the privileged mode capability). Any other user can restore any tape file in his log-on account if he has save-access to the group to which the file belongs (but cannot restore those with negative file codes unless he also has the privileged mode capability). If the file on tape is protected by a lockword, the lockword must be supplied in the :RESTORE command. (Users logged on interactively are prompted for omitted lockwords.) However, System Managers, Account Managers (restoring within their own account), and System Supervisors are not required to provide passwords.

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). The format of this listing, and the interpretation of the error messages that may occur, are given in the text discussion.

EXAMPLE

To retrieve from the file named BACKUP all files belonging to the user's log-on group, the user enters:

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

If a disc file satisfying the @ specification already exists in the system, it is not restored, since the KEEP option was specified.

TEXT DISCUSSION

Page 6-76.

: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

NOTES

When logging resumes, a special Log Record is emitted (provided type 0 is enabled) 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.

EXAMPLE

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

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

The System Supervisor determines that error 46 is a recoverable error. He logs onto the system in session mode at any terminal and corrects the problem. He then enters:

```
:RESUMELOG
```

In response, the following messages appear at the system console:

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

TEXT DISCUSSION

Page 6-84.

Displays the current Log File's number and the amount (percentage) of available file 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

NOTES

The information displayed appears in this format:

LOGFILE LOG*xxxx* IS *yy*% FULL

xxxx Log File Number
yy 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 (described in the text discussion), both messages appear.

EXAMPLE

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

:SHOWLOG

The following display appears:

LOG FILE LOG0655 IS 2% FULL

TEXT DISCUSSION

Page 6-84.

:SHOWQ

Displays information about the scheduling of processes.

SYNTAX

:SHOWQ

PARAMETERS

None

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes; aborts
Capability required:		OP

NOTES

If this command is entered at the system console, the queues are displayed directly on the console. If entered within a session or job, the command requires the user to have system supervisor capability and the output is transmitted to the standard listing device ($STDLIST).

A sample listing and explanation are given in Appendix D.

TEXT DISCUSSION

Page D-10.

Stores onto magnetic tape any disc file or fileset.

SYNTAX

:STORE [*filesetlist*] ;*tapefile* [;SHOW] [;FILES=*maxfiles*]

PARAMETERS

filesetlist A list of one or more files or filesets to be copied, written in this format:

[*fileset*[,*fileset*] . . .]

NOTE

The *filesetlist* parameter has the following limits, depending on the form of the parameter.

If the form @.@.@, meaning copy all files in the system, is used, the limit, of course, is 1.

Up to ten accounts can be specified by name with the form @.@.*accountname*. All files in all ten accounts will be copied. Up to 15 groups within an account can be specified by name with the form @.@.*groupname.accountname*. All files in all 15 groups will be copied.

Up to 20 files can be specified by name within a group in an account with the form *filename.groupname.accountname*. All 20 files will be copied.

In this list, *fileset* must be written as one of the combinations noted below. If the entire *filesetlist* parameter is omitted, the default value is @ (all files of the log-on group are copied). If any file belonging to a fileset requires a lockword, you may supply that lockword. If you fail to supply it (while programming in batch mode), the file will not be stored. If you fail to supply it (in interactive mode), you are prompted for the lockword. (Optional parameter.)

tapefile The name of the destination tape file onto which the stored files are written. This can be any magnetic tape file from the output set. This file must be referenced in the **formaldesignator* format. (Required parameter.)

SHOW A request that the names of the stored files be listed. If SHOW is omitted, only the total number of files stored, the names of the files not stored, and the number of files not stored, are listed. (Optional parameter.)

maxfiles The maximum number of files that may be stored. If omitted, the number 4000 is used by default. (Optional parameter.)

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	Yes
	programmatically?	Yes
Breakable?		Yes; aborts
Capability required:		See below

NOTES

A user with the System Manager or System Supervisor capability can dump any user file in the system. An account-manager user can dump any file in his account (but cannot dump those with negative file codes unless he also has the Privileged Mode capability).

The *fileset* parameter allows you to request dumping of one file alone, or various sets of files. It contains three positional-fields separated by periods: the file, group, and account fields. The file field permits you to indicate a specific file or all files within the units designated by the other fields. The group field denotes the group to which the files belong. This can be your log-on group, any other group, or all groups within the accounts specified by the account field. The account field indicates the account or accounts to which the groups belong. This can be the log-on account, any other account, or all accounts in the system. (To specify *all* files, groups, or accounts, you enter the character @ in the appropriate field. The omission of an entry in the group or account field denotes the log-on group or account.) For the *fileset* parameter, the combinations of entries shown below are possible:

File Field	Group Field	Account Field	Entry Example	Meaning
filename	*groupname*	*accountname*	FILE.GROUP.ACCT	The file named, in the group and account designated.
filename	*groupname*		FILE.GROUP	The file named, in the group designated under the log-on account.
filename			FILE	The file named, under the log-on group.
@	*groupname*	*accountname*	@.GROUP.ACCT	All files in the group named, under the designated account.
@	*groupname*		@.GROUP	All files in the group named, under the log-on account.
@			@	All files in the log-on group.
@	@	*accountname*	@.@.ACCT	All files in all groups under the account named.
@	@		@.@	All files in all groups under the log-on account.
@	@	@	@.@.@	All user files in the system.

Before issuing a :STORE command, tapefile must be identified as a magnetic tape file. This is done through the :FILE command. In this case, the :FILE command should be written in the following format, including no parameters other than those shown:

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

The device parameter must be the device class name or logical unit number of a magnetic tape unit.

All other parameters for tapefile are supplied by the :STORE command executor; if you attempt to supply any of these, the :STORE command is rejected.

The format of the listing, and the interpretation of the error messages that may occur, are given in the text discussion.

If the break key is pressed during the store operation, the operation stops after storing the current file, and further output is suppressed.

EXAMPLES

To copy all files in the group GR4X (in the user's log-on account) to a tape file named BACKUP, the following commands are entered. A listing of the files copied appears on the standard list device.

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

NOTE

The system will ask the operator for the device number of tape named BACKUP.

Explicit or implicit redundancies are permitted in *filesetlist*, but once a file has been locked down, any subsequent reference to it in *filesetlist* results in the message BUSY even though execution of the :STORE command continues.

Suppose the file identified as FN.GN.AN is a member of the fileset referenced by @ in the following command.

```
:STORE @, FN.GN.AN; *DUMPTAPE; SHOW
```

The command is executed successfully, but the fileset# and msg notations under FILES NOT STORED on the listing will show:

filename	groupname	acctname	fileset#	msg
FN	GN	AN	2	BUSY

This same file will also be noted under FILES STORED.

TEXT DISCUSSION

Page 4-7, 6-72.

: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

NOTES

When this command is entered, the current Log File is closed, a new Log File is created and opened, and the new Log File becomes the current Log File.

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

The information displayed appears in this format:

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

where:

xxxx	Previous log file number
yy	Percentage of file space used
zzzz	Current log file now opened (increment of 1)

If logging is enabled but currently suspended, MPE outputs the message

NO LOGGING

EXAMPLE

To switch logging to a new file, the System Supervisor enters:

```
:SWITCHLOG
```

The following display appears:

LOG FILE LOG2015 IS 72% FULL
LOG FILE NUMBER 2016 ON

TEXT DISCUSSION

Page 6-82.

:SYSDUMP

Starts Configurator dialogue and copies MPE to tape.

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.* The formal file designator used by MPE is DUMPTAPE. The file open/close (FOPEN/FCLOSE) options provided by the :SYSDUMP command executor are:

NEW; REC=1024,,U,BINARY; SAVE; ACC=OUT

All other file characteristics are supplied by MPE default. Back-referencing, formal file designators, file characteristics, and default options are all explained in the discussion of the :FILE command in the *MPE Commands Reference Manual.* (Required parameter.)

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

NEW; REC=,,V,ASCII; CCTL; ACC=OUT

If the *auxlistfile* parameter is omitted, the session list device (typically the user's terminal) is assigned by default. (Optional parameter.)

USE

Available	in Session?	Yes
	in Job?	Yes
	in Break?	No
	programmatically?	No
Breakable?		Yes; suspends
Capability required:		OP

NOTES

In addition to the names of the files to be used for the *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. Then, in the :SYSDUMP command, you precede each filename with an asterisk to indicate a back-reference to the definitions in the :FILE commands.

EXAMPLE

To begin dialogue with the MPE Configurator, the :SYSDUMP command is entered. At the end of the dialogue, the MPE configuration requested is copied to the file named DUMP. Any listings requested during the dialogue 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 these 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 6-3.

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 User creates an account, he defines for it a capability set that determines whether or not users communicating with MPE through this account can be allowed certain functions. When an Account-Manager User defines the users of his account, he associates with each user an individual capability set that may allow the 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 individual user — it defines the extent to which he must understand and interrelate with MPE, and permits a user to ignore aspects of MPE that do not apply to him.

Capability sets are also defined for groups by the Account-Manager User. 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, 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 the customer. He, 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, users for record purposes.

- List file attributes.
- Obtain account reports.
- Store and restore any or all files on the system.

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

SYSTEM SUPERVISOR ATTRIBUTE (OP). Allows the user to have day-to-day external control of this system. It allows him to manage scheduling subqueues, alter the system configuration, maintain the system logging facility, and display various items of system information. This attribute may be assigned by a user with the System Manager Attribute.

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.

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.

GROUP LIBRARIAN ATTRIBUTE (GL). Similar to the Account Librarian Attribute, but limits the special file-access modes allowed the user to his home group. This attribute is assigned by users with the Account Manager Attribute. It could be used, for example, where it is desired that only one user can have the capability to alter files within a particular group. This user could be assigned the Group Librarian Attribute and his access modes could be made greater than those of other users.

DIAGNOSTICIAN ATTRIBUTE (DI). Permits the user to run certain device and cpu verification programs. Normally, the Hewlett-Packard service representative (Customer Engineer) is the only user who may need this attribute. The diagnostician attribute is assigned by users with the Account Manager attribute.

STANDARD USER. All other users not specifically assigned one or more of the above user attributes fall into the class of Standard User by default.

FILE ACCESS ATTRIBUTES

The file-access attributes determine whether the user has the capability to:

- Save user files permanently (SF). This means that those users who do not have this capability cannot use the :BUILD, :SAVE, and :RESTORE commands (described in the *MPE Commands Reference Manual*). Nor can they use the FOPEN and FCLOSE (permanent) intrinsics (described in the *MPE Intrinsics Reference Manual*). Users who do not have this capability can, however, open job/session temporary files, but these will automatically be deleted at termination time.

- Use non-sharable devices (ND). This attribute is needed to use real devices other than disc. The user will have complete and absolute control of the device. Some examples are card readers, line printers, magnetic tape units, plotters, etc. It should be noted that this attribute is not necessary for the standard job/session input and listing devices.

These two attributes can be assigned in any combination by the Account Manager user.

CAPABILITY-CLASS AND PROGRAM ATTRIBUTES

These attributes define the general resources available to a user and the programs he writes and runs.

Each time someone runs a program, MPE automatically assigns that program the user and file-access attributes of that user. But the capability-class attributes assigned to the program are designated by the user who originally prepares the program; they are passed to the MPE Segmenter as parameters of the command that prepares the program. If the preparing user does not designate capability-class attributes for his program, MPE assigns, by default, the standard capabilities possessed by that user — interactive access (IA), batch access (BA), or both. (When programs prepared from passed files or job temporary files are run, they are assigned the standard capabilities (interactive and/or batch access) possessed by the user who runs them.)

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.*

WARNING

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 cannot be responsible for system integrity when programs written by users operate in the privileged mode.

- 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.

- 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 use the remote job entry features of the system.

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

The local attribute is a double-word quantity (i.e., two 16-bit computer words) used only for special applications that require further unique classification of users by other users. MPE does not reference or make use of this attribute in any way. It is defined by System or Account Managers and is used by accounts or groups for any purpose desired. You may access the local attribute in your program via the WHO intrinsic described in the *MPE Intrinsics Reference Manual.*

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 (see figure 3-1) 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.

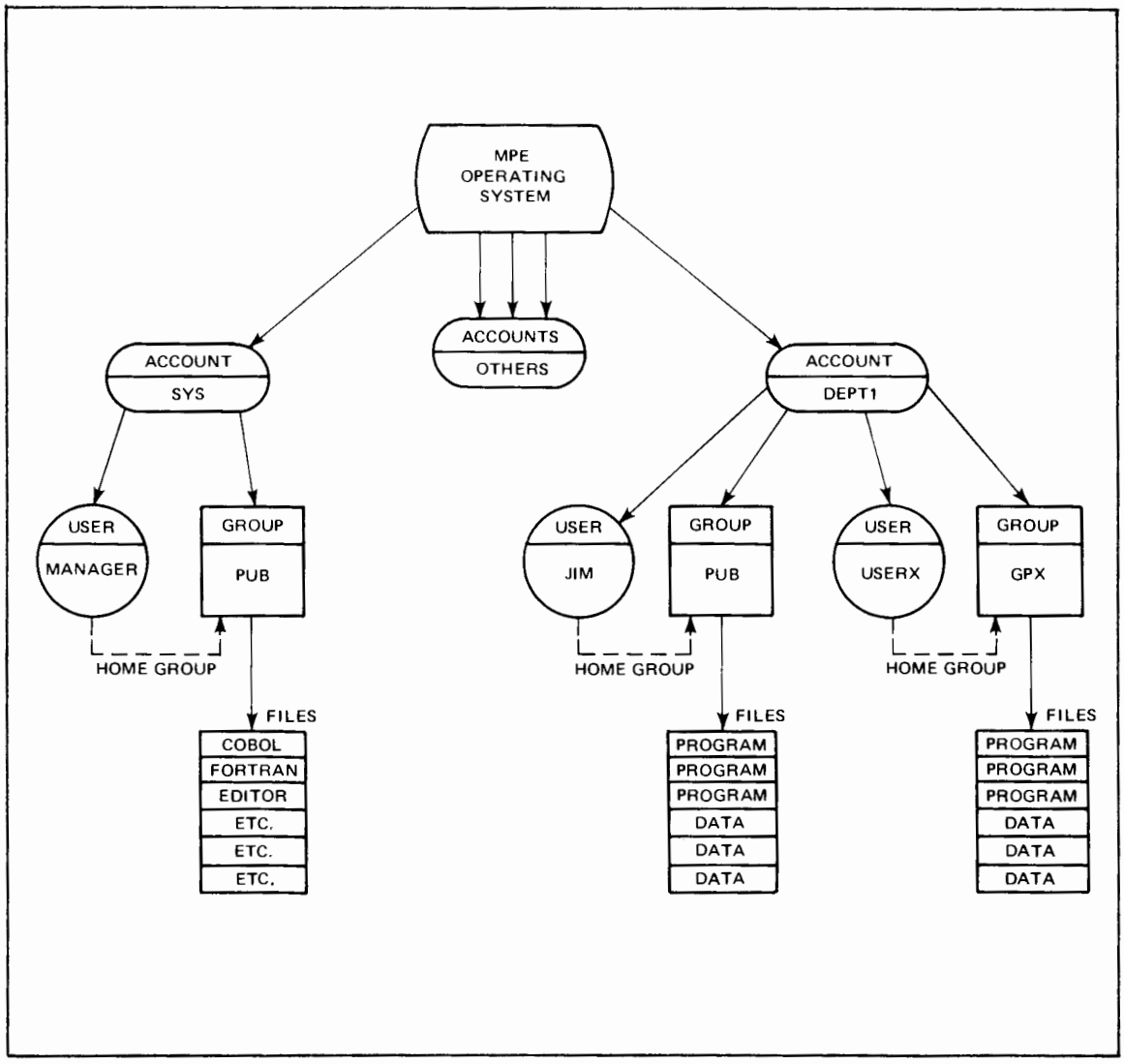


Figure 3-1. Account/Group/User Organization

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 3-2.)

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.

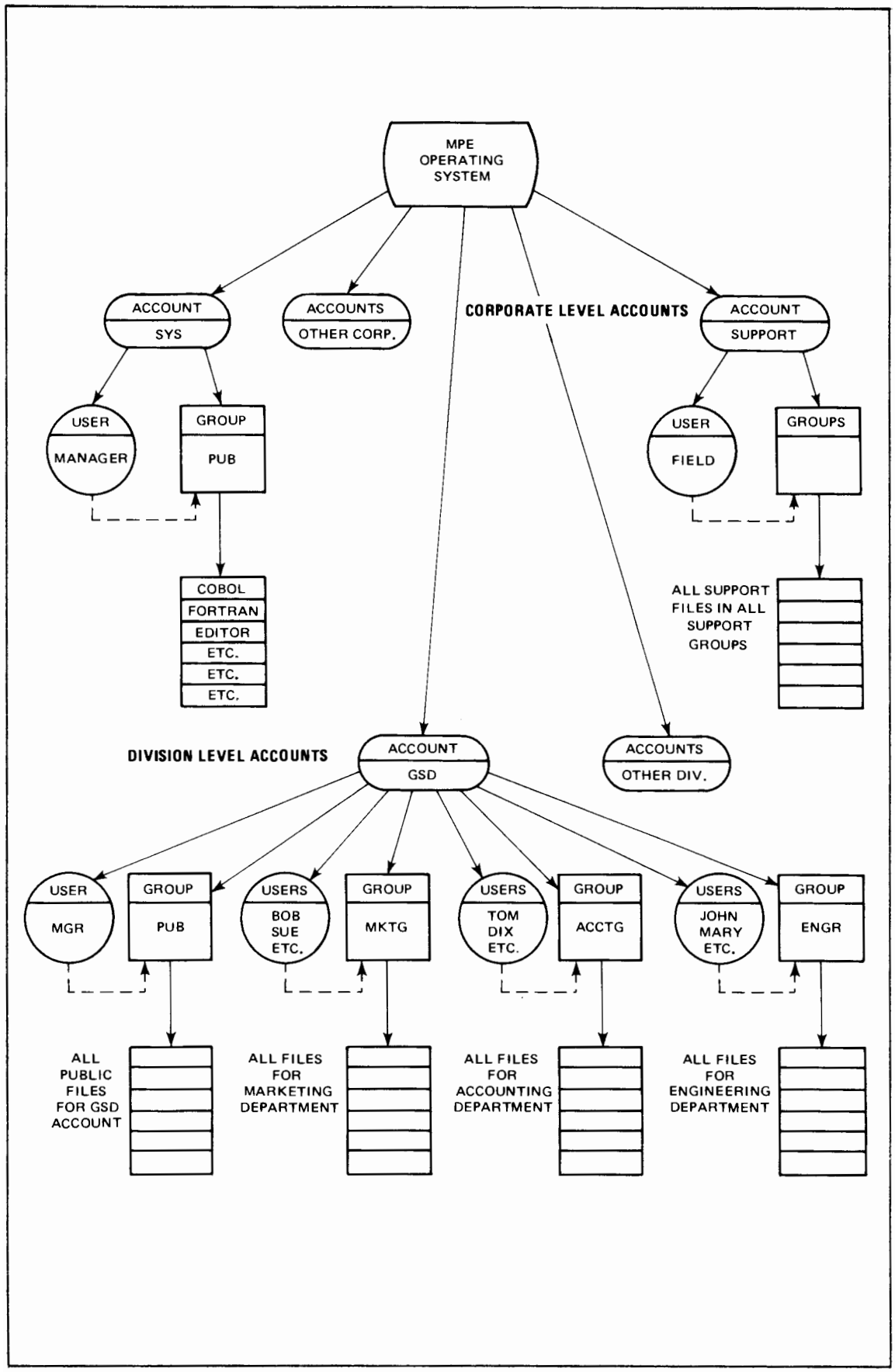


Figure 3-2. Accounting Structure in a Corporation

- 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. (Notice that these provisions apply to disc files only.) These restrictions are based on two 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 instance) to whom the access modes specified are permitted.

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.

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.

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.

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 that user.

At the account level, five access modes are recognized:

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

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.

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 are 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 any or all files on the system.

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 — SM, AM, SF, ND, PH, DS, MR, PM, CS, IA, BA (These capabilities are defined in Section II under the :NEWACCT command.)

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. Spooling will shut down temporarily if the process cannot obtain enough disc space to satisfy users of the spooling facility. 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.

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. However, it is possible that if the limits are exceeded, a System Manager would be denied access to the system. In this case, the System Manager would have to log onto the system using the optional log-on parameter HIPRI (discussed in the *MPE Commands Reference Manual*).

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 DISPATCH, 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. If there are user programs which must reside in the PUB group of the SYS account, the programmer could make use of the local attribute in his program (by calling the WHO intrinsic, described in the *MPE Intrinsics Reference Manual*) to limit access of the program to SYS account users only.

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, GA

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, LISTDIR, which will format all of the octal information that is printed by the commands. Its use 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. 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*. For any particular resource, the sum of all group counts within any account always equals the total count accumulated by the account.

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 AND RESTORING 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:

```
ACCT DIAG      TO BE PURGED?
```

The System Manager responds with a YES answer. At this time, all files, all users, and all groups 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 or execution limits such as cpu or connect time limits.

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.

ACCOUNT MANAGER CAPABILITIES

SECTION V

When an account is created by a System Manager, it invariably 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 will be the *Account Manager* and will have the Account Manager (AM) capability attribute. This attribute allows the user certain expanded capabilities over other users within a specific account. The Account Manager user has less capability than the System Manager but does have the final control of the overall use of his particular account. He 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; LOCATTR = 184
```

Note that the System Manager created the account with one optional parameter. Therefore, the newly created SMITH account and its attributes are the defaults assigned by the operating system, except local attribute 184. The account structure is as follows:

Account	
acctname	SMITH
password	Null (none)
filespace	Unlimited
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	Same as account files
cpu	Same as account time
connect	Same as account time
capabilitylist	IA, BA
fileaccess	(R, X: ANY; A, W, L, S: AL, GU)
User	
username	TOM
password	Null (none)
subqueuename	Same as account *subqueuename*
localattribute	Same as account *localattribute*
homegroup	PUB
capabilitylist	Same as account capabilities

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. TOM, as an Account Manager user, 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.

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

```
:HELLO TOM.SMITH
```

He could log onto the system in the :JOB mode if he desires, but in these examples we will use the session mode for convenience. As can be seem, initially there is no password on this account (because the System Manager did not assign one), or the group, or the Account Manager user. The Account Manager can define new groups and users in order to divide the account logically. In this regard, the Account Manager may assign as many groups and users as he likes; there is no logical limit to the number of groups or users.

The limitation to the number of users and groups 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 entire 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 while the operating system is online.

The Account Manager user has the following commands for his exclusive use:

```
:ALTGROUP
:ALTUSER
:LISTACCT        (For his account only)
:LISTGROUP
:LISTUSER
:NEWGROUP
:NEWUSER
:PURGEGROUP
:PURGEUSER
:REPORT          (For his account only)
```

The above commands, which are the domain of the Account Manager, will be discussed in detail in this section. Continuing related examples will be used to illustrate typical situations.

DEFINING GROUPS

To create a new group, the Account Manager user 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	
groupname	JOHNSON
filespace	10000 sectors
(*)	
groupname	MASON
filespace	10000 sectors
(*)	
groupname	BLAKE
filespace	10000 sectors
(*)	
groupname	PAYROLL
filespace	Unlimited
password	MANYBUKS
capabilitylist	BA
(*)	
Users	
username	JIM
password	J2DEV
homegroup	JOHNSON
(*)	
username	MARY
password	AQUARIUS
homegroup	MASON
(*)	
username	BOB
password	ME98
homegroup	BLAKE
localattribute	184
(*)	

*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.

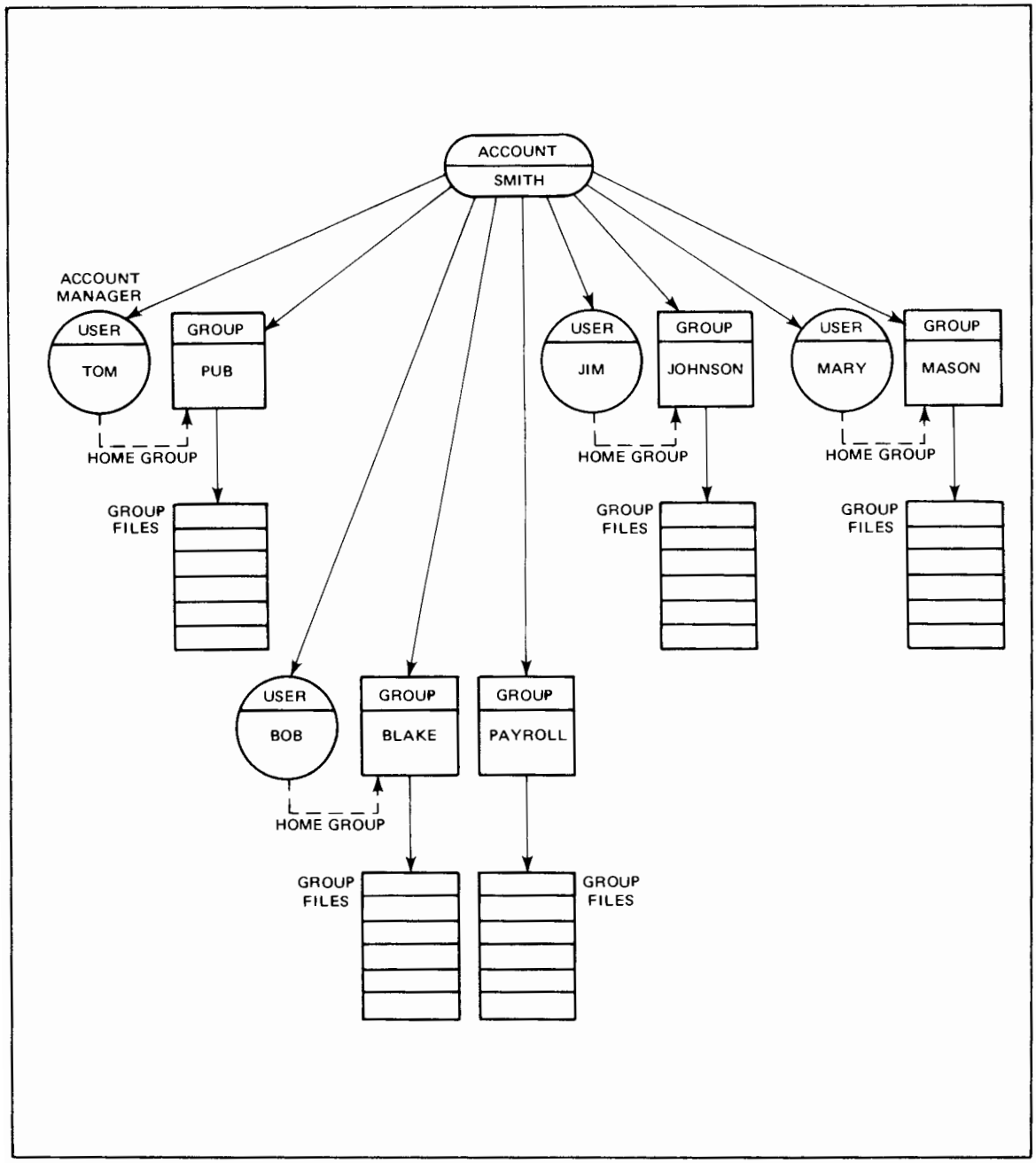


Figure 5-1. Sample Account Structure

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.

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.

The :PURGEGROUP command can be useful in another way. Suppose Bob Blake has utilized 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 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
ERR 119
```

What has taken place is the removal of all of Bob Blake's files from within his group. However, his group structure still exists since the Account Manager was logged onto the system in the home group BLAKE at the time the command was issued. This is the desired effect, and the error indication can be ignored. This use of the :PURGEGROUP command is useful for removing all files within a group, especially if there are a great many files to be removed.

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 uses the FCOPY utility (desired 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 issues the command as follows:

```
:PURGEGROUP JOHNSON
GROUP JOHNSON TO BE PURGED? 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
:
```

At this time the user name JIM and all its attributes will be removed from the account structure. Note that, like the :PURGEGROUP command, if someone had been logged onto the system with the user JIM under the account SMITH, that user would have prevented the user name Jim from being removed. The Account Manager would have had to wait until the person using that user name had logged off the system before he could try the command again.

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

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. An example for the System Manager is detailed in Section IV.

SYSTEM SUPERVISOR CAPABILITIES

SECTION VI

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

- Creation of magnetic tapes of all user files on the system for offline storage.
- Creation of magnetic tapes for backing-up and/or modifying the system.
- Displaying certain system information for the purpose of determining the optimal performance of the system.
- Permanently allocating/deallocating programs in virtual memory.
- Exercising scheduling control over processes in order to obtain the best system performance.
- Managing the system log files through the logging facility.

The System Supervisor (OP) capability differs from the System Manager (SM) capability (which is mostly concerned with the accounting system) in that the user with the OP capability 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 the functions as System Supervisor, the user with the OP capability has the following commands for his exclusive use:

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

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

SYSTEM CHANGES AND FILE STORAGE

Hewlett-Packard furnishes MPE as a set of prepared program files on magnetic tape. 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 of the capabilities of the MANAGER user except the System Manager (SM) capability. This user is identified by the user name FIELD.

- Two system accounts. The accounts are SYS and SUPPORT. The MANAGER user is associated with the SYS account. 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 HPXXXXX where XXXXX is the Hewlett-Packard product number of the appropriate software module.

In addition, magnetic tapes 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.

Once the system is started (by the Console Operator) and some accounting structure and files exist, the user with System Supervisor capability can call the MPE Configurator program :SYSDUMP to:

1. Perform a system and file back-up to magnetic tape by:

   a. Copying the operating system only.

   b. Copying the operating system plus accounting structure.

   c. Copying the operating system, accounting structure and all files that have been altered since a given date.

   d. Copy 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

A duplicate of the running system on magnetic tape is the only way that you can provide back-up. If no such tape 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 back-up of the system and of all the files on the system be accomplished 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.

The :SYSDUMP command format is

:SYSDUMP *dumpfile* [,*auxlistfile*]

where

dumpfile The actual file designator of the output file (device) on which the system is to be copied. The format for writing actual file designators is explained in the *MPE Commands Reference Manual*. Typically, in this case, the designator is a file name containing up to eight alphanumeric characters, beginning with a letter. This must indicate a magnetic tape file. The formal designator is DUMPTAPE. The FOPEN/FCLOSE options provided by the executor are: NEW; REC = 1024,, U, BINARY; SAVE; ACC = OUT. (Required parameter.)

auxlistfile The actual designator from the output set, specifying an ASCII file which contains all listings requested during the Configurator/User dialogue, and the information concerning files dumped and not dumped during the backup portion. The default is $STDLIST.

The formal designators used by SYSDUMP for *auxlistfile* are SYSLIST and SEGLIST (the latter is used by the Segmenter process). The FOPEN/FCLOSE options provided by the executor are NEW; REC = −72,, V, ASCII; CCTL; ACC = OUT. (Optional parameter.)

In addition to the names of the files to be used for the *dumpfile* and optional *auxlistfile* output, you must specify what type of devices these files are to reside on. This is done by preceding the :SYSDUMP command with MPE :FILE commands defining each of the two files. Then, in the :SYSDUMP command, each filename is preceded with an asterisk to indicate back referencing. :FILE commands are discussed in the *MPE Commands Reference Manual*.

As an example, to begin dialogue with the MPE Configurator, the :SYSDUMP command shown below is entered. At the end of the dialogue, the MPE configuration, together with the current directory and accounting information and all user files specified, is copied to the file named DUMP. (The *auxlistfile* parameter is not used.) The :FILE command specified that DUMP is a magnetic tape file. (In this command, the TAPE parameter is a device class name defined during a previous configuration.)

```
:FILE DUMP; DEV=TAPE
:SYSDUMP *DUMP
ANY CHANGES? NO
ENTER DUMP DATE? 0
LIST FILES DUMPED? NO
```

The tape is now written. When it is complete, a count of files dumped, a count of files not dumped, and a list of files not dumped will be printed on the terminal.

When using the Configurator (:SYSDUMP) to modify the operating system, the user requires certain background information. The information he needs depends on the types of changes he wishes to make, as noted in this section. Changes are made through an interactive dialogue in the session mode between the user and the Configurator; the step number of the dialogue

where the change is made is noted in the heading that pertains to the change. Recommended parameter settings for configuration dialogue are discussed in this section and are given in Appendix C.

NOTE

It is recommended that all changes to the system segmented library SL.PUB.SYS be made only through the use of the Configurator program. The Configurator program copies the library to a temporary file where it can be changed conveniently.

CHANGING MAIN MEMORY SIZE [2]

MPE runs on HP 3000 Computers with main-memory of the following sizes:

64K (K = 1024 words)	128K	224K
96K	160K	256K
	192K	

The size specified by the user during configuration should be the actual size of main-memory delivered with the machine on which this MPE configuration is to run. This entry is required so that other configuration parameters (such as table sizes) that depend on main-memory size can be set up correctly.

The system maintains a special file (CONFDATA) located in the PUB group of the SYS account. This CONFiguration DATA file maintains a separate record of key information for each configurable main memory size. The Configuration program and the Initialization (start-up) program will access only the record that matches the main memory size specified by default or by the user. Unauthorized user access to the CONFDATA file could damage the system.

CHANGING INPUT/OUTPUT DEVICE CONFIGURATION [3]

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. 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 Name
1	Disc
4	Magnetic Tape
152	Terminal

  Normally, you would assign logical devices in ascending order. As the above example illustrates, you can assign any arbitrary number within the ranges specified.

- 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 DRT is located in fixed-memory locations beginning at octal address 14. The maximum upper limit for the DRT is location 777_8, thus limiting the maximum number of four-word DRT entries to 125 (decimal) entries. This also limits the number of physical device controllers to 125.

  Since each DRT entry is always four words long, it is convenient for the hardware to map controllers to DRT addresses simply by multiplying by four. Since the DRT begins at location 14, the lowest controller (DRT entry) number is 3 ($14_8/4_8 = 3_8$). DRT entry numbers 0, 1, and 2 do not exist. However, the memory locations octal 0 through octal 13, where DRT 0, 1, and 2 would normally be located, are used by the operating system.

  When re-configuring the system, you need to know the highest DRT entry number that can be assigned to a device. You determine this by adding three to the total number of controllers planned. This value may not exceed 127.

  You also need to know the DRT entry number of any device to be added or deleted. This is a hardware-dependent value, ranging from 3 to 127. It is determined by a set of jumpers on the device controller board.

  Since the DRT is a main memory table, it is not advisable to include more than two spare (unused) entries. This will keep the table size to the minimum required for efficient utilization of main memory.

- Unit Numbers

  When a controller services only one device, that device is generally 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.

- Channel Number

  The input/output (I/O) system allows multiple controllers on a single selector channel. The system will transfer data on only one controller per 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 channel, even though physically they need not be on the same channel to share the same 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.

- Device Type

  This number determines the type of device, where

OCTAL	DECIMAL
0	0 = Moving-Head Disc
1	1 = Fixed-Head Disc
10	8 = Card Reader
11	9 = Paper Tape Reader
20	16 = Terminal
22	18 = Synchronous Single-Line Controller
23	19 = Hardwired Serial Interface
24	20 = Printing Reader/Punch
27	23 = Programmable Controller
30	24 = Magnetic Tape
40	32 = Line Printer
41	33 = Card Punch
42	34 = Paper Tape Punch
43	35 = Plotter

- Device Sub-Type

  This characteristic is specified for a device whose software driver handles similar devices with different characteristics (such as hard-wired terminals versus terminals connected through a modem). It is a number ranging from 0 to 15, depending on the actual device referenced (as specified in Appendix A).

- Terminal Type

  If the device is a terminal (device type 16) which is hardwired (subtype 0), a default terminal type can be configured. This number, which can be overridden by the TERM= parameter in the log-on command, can be one of the following:

  0 HP 30124A (HP 2749B), ASR-33, or ASR-35
  1 ASR-37
  3 Execuport 300

4 HP 30123A (HP 2600A), or Datapoint 3300
5 Memorex 1240
6 HP 30120A (HP 2762A), GE Terminet 300 or GE Terminet 1200
9 HP 30122A (HP 2615A, Mini-Bee)
10 HP 2640A, HP 2644A (Character mode)
11 HP 2640A, HP 2644A (Block/line mode)

- Option to Accept Job and Session Input Stream (J)

  You can optionally specify that this device can accept an input stream from a job or session. In other words, it 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. When the device is additionally 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 dialogue 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 duplicative if all input operations are 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 this 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, a user can see what has been input to the system software.

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/;) 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.

- 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 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.

While it is true that any number of device classes can be assigned to a device, or none at all, except for a minimum class of DISC, some level of restraint should be used. Device classes are kept in the Logical Device Class Table. This table, known to the operating system, can become extremely large when an inordinate number of classes are assigned to devices. The table then requires more system overhead time for swapping between main memory and virtual memory whenever files are opened.

MPE defines five 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 five 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)
0-7	Direct access (DA)
8-15	Serial input (IN)
16-23	Concurrent input/output (I/O,C)
24-31	Non-concurrent input/output (I/O,NC)
32-39	Serial output (OUT)

The distinction between concurrent and non-concurrent is as follows:

concurrent — both input and output operations are allowed while the file is open. For example, terminals may be written to or read from in any order.

non-concurrent — operations are confined to either input or output while the file is open, never a mixture. For example, a magnetic tape is written to or read from during any one file-open, but the two are not logically done to the same tape at the same time.

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 direct access DATs is automatically assigned a DCAT of direct access. For example, class DISC is assigned direct access (DA) DCAT.

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.

CHANGING SYSTEM TABLES AND QUEUE PARAMETERS [4]

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 four 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 four 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.

  d. Memory Management Table (MTAB)
  This table contains various information formats in five-word entries. It is main memory resident. 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 particular 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. 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 two 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.

- The number of buffers

  a. Terminal Buffers in the system
  Terminal buffers are main memory resident. Each buffer is 18 words in length (36 characters). Set the number of buffers to about 5 times the actual number of terminals on the system.

b. System Buffers in the system
System buffers are main memory resident. Each system buffer entry is 128 words in length. System buffers are used by the file management system and the input/output system to reduce the number of extra data segments required for any particular user. Experimentation with the number of entries may be done.

- 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 list 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 a main memory resident table. Each entry is four words long. 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 12.

- The contents of the message catalog

  This is the system data segment where MPE keeps its standard messages and error messages. It is a special file made by the Configurator program (:SYSDUMP). The source file is CATALOGS in a group HP 32002 of the SUPPORT account.

  Changes to the catalog and its source file are made periodically by Hewlett-Packard. Normally, the next sequential message numbers are used, in ascending order. If you desire to put messages into the catalog, it is recommended that you block-fill the catalog with a number of dummy message numbers before adding your own messages.

  When a change is made to the catalog, the entire catalog must be input along with the change. See Appendix G for format and methods of making changes.

CHANGING MISCELLANEOUS VALUES RELATING TO LOG-ON TIME, RIN'S AND JOBS ALLOWED IN SYSTEM AND EXECUTION [5]

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.
- The default central processor time limit for jobs.

CHANGING LOGGING CHARACTERISTICS [6]

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 records in the log file.
- The size of the log file.

CHANGING DISC ALLOCATION [7]

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 [8]

All processes competing for access to the central processor access it through the MPE Master Scheduling Queue. (Scheduling queues are described later in this section.) You can change the time quantum, in milliseconds, allowed the time-sharing class (CS and DS). The priority ranges for this class also may be changed. Experimentation with the priorities can improve system performance. It also can degrade system performance. Changing the priorities should be done very cautiously. Note that 150 is highest priority and 250 is lowest. Also, the three priorities must be monotonically increasing.

CHANGING SEGMENT LIMITS [9]

You can change the limits on code and data segments, as follows:

1. 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.

2. Maximum number of code segments per process.

3. 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.

4. 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 3 above.)

5. Maximum number of data segments per process.

6. 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 recommend 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 3 above). In any case, the minimum recommended for any configuration is 800 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 [10] AND SEGMENTS IN THE SYSTEM LIBRARY [11]

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 user with System Supervisor (OP) capability can either re-configure or copy his running MPE system onto magnetic tape by entering the :SYSDUMP command through his terminal

during a session. This command can be entered at any time. However, prior to entering the command, file references to the magnetic tape (for output) must be made. A file reference to a list device can also be made (optional).

When the :SYSDUMP command is entered, the Configurator begins an interactive dialogue with the user by asking if he wants to make any configuration changes. If he responds NO, the Configurator copies the running system to tape. If he responds YES, the Configurator continues its dialogue with the user.

The Configurator's output consists of questions (ended by a question mark) and statements (ended by a period). The content of the questions generally indicates the type of answer required. To those questions requiring a simple positive or negative answer, the user responds with YES (or simply Y) or NO (or N, or simply a carriage return). Other questions contain values followed by a question mark; they normally quote an existing parameter value and ask whether the user wants to change it. To retain the quoted value, the user enters a carriage return. To change the value, he enters the new value desired. In any case, the user must always conclude an entry with a carriage return to transmit the entry to MPE.

CONFIGURATOR/USER DIALOGUE

SYSTEM MODIFICATION

The Configurator/User dialogue proceeds as follows, with Configurator (:SYSDUMP) output shown verbatim in uppercase letters, and user input shown in uppercase letters, underlined. This example is primarily intended for users desiring to make changes to the operating system. Later examples in this section will deal with system back-up and offline file storage only. Prior to entering the dialogue, the user logs onto the system and inputs *at least* a file reference to a magnetic tape.

A typical example is:

:FILE SYSMOD; DEV=TAPE
:SYSDUMP *SYSMOD

(Assume that device class TAPE is a magnetic tape.)

The dialogue between Configurator and user then commences as follows:

Step No.	Dialogue
0	ANY CHANGES? To prepare for changes, enter YES To omit changes, and skip to Step 12, enter NO.
1	SYSTEM ID = HP 32002 V.<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

Step No. **Dialogue**

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.

2 MEMORY SIZE = <XXX>.?

In this message, 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: 64, 96, 128, 160, 192, 224, or 256. This denotes memory size in a multiple of 1024 words. To retain the present memory size, enter a carriage return.

3 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 4, enter NO.

3.1 LIST I/O DEVICES?

To print a list of input/output devices currently assigned to the system, enter YES. The format of the output is:

```
LOG   DRT   UNIT   CHAN   TYPE   SUB-   TERM   REC     OUTPUT   MODE   DRIVER   DEVICE
DEV    #      #                   TYPE   TYPE   WIDTH    DEV            NAME    CLASSES
```

To suppress this listing, enter NO.

NOTE

The prompt in Step 3.1.1, below, appears only if a synchronous single-line controller (SSLC) was previously configured into the system.

3.1.1 LIST CS DEVICES

To print a list of the characteristics of all SSLCs 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
```

In this format, the headings denote the following information:

HEADING	MEANING
LDN	Logical device number.
PM	Reserved for future use; presently always 0.
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.

To suppress this listing, enter NO.

3.2 HIGHEST DRT=<XX>.?

In the output, 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.

3.3 LOGICAL DEVICE #?

To specify a device to be added or removed, enter the logical device number of that device.

To skip to Step 3.4, enter zero or a carriage return.

Step No.	Dialogue

3.3.1 DRT #?

To add a device, enter its DRT entry number.

To remove a device and return to Step 3.3, enter zero.

3.3.2 UNIT #?

Enter the physical hardware unit number of the device, if the device shares its controller with other devices.

Otherwise, enter zero to continue.

3.3.3 CHANNEL #?

If the device is to be on a multicontroller channel, enter channel number; if not, enter zero.

3.3.4 TYPE?

Enter the device type, where

Octal	Decimal
0	0 = Moving-Head Disc
1	1 = Fixed-Head Disc
10	8 = Card Reader
11	9 = Paper Tape Reader
20	16 = Terminal
22	18 = Synchronous Single-Line Controller
23	19 = Hardwired Serial Interface
24	20 = Printing Reader/Punch
27	23 = Programmable Controller
30	24 = Magnetic Tape
40	32 = Line Printer
41	33 = Card Punch
42	34 = Paper Tape Punch
43	35 = Plotter

3.3.5 SUBTYPE?

Enter the device sub-type in the range 0 to 15. (Appendix A)

3.3.6 TERM TYPE?

This question is asked only if type is 16. To specify a default terminal type to be used at log-on, enter a number as follows:

0 HP 30124A (HP 2749B), ASR33 or ASR35
1 ASR37

Step No. **Dialogue**

2 (Reserved)
3 Execuport 300
4 HP 30123 (HP 2600A), or Datapoint 3300
5 Memorex 1240
6 HP 30120A (HP 2762A), GE Terminet 300 or 1200
7 (Reserved)
8 (Reserved)
9 HP 30122A (HP 2615A) MINI BEE
10 HP 2640A, HP 2644A (Character Mode)
11 HP 2640A, HP 2644A (Block/line mode)

NOTE

For Distributed Systems Dummy terminals, or if the term type is not known, enter a carriage return.

NOTE

The prompts in Steps 3.3.6.1 through 3.3.6.24, below, appear only if you are configuring an SSLC. If you are *not* configuring an SSLC, the dialogue skips to Step 3.3.7.

3.3.6.1 PROTOCOL?

To define the SSLC protocol, enter 1.
(The Hardwired Serial Interface is PC1.)

3.3.6.2 LOCAL MODE?

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). (Enter 1 for the Hardwired Serial Interface.)

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 HP 2780/3780 Emulator is usually a primary station.

3.3.6.3 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

Step No.	Dialogue

3.3.6.4 RECEIVE TIMEOUT?

Enter the positive number of seconds the Emulator will wait to receive text before terminating the read mode. Entering a carriage return provides a 20-second timeout.

NOTE

For all timeout responses: 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.

3.3.6.5 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.

3.3.6.6 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

Steps 3.3.6.7 through 3.3.6.9 apply only to SSLCs with switched lines *not* connected via modems. If the SSLC is connected through a modem, the dialogue skips to Step 3.3.6.10. If the SSLC is hardwired, the dialogue skips to Step 3.3.6.12.

3.3.6.7 DIAL FACILITY?

Enter YES when calls can be dialed from the local station. Enter NO when they cannot. (Enter NO for the Hardwired Serial Interface.)

3.3.6.8 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. (Enter NO for the Hardwired Serial Interface.)

3.3.6.9 AUTOMATIC ANSWER?

Enter YES if the local modem can automatically answer calls. Enter NO if manual answering is required. (Enter NO for the Hardwired Serial Interface.)

Dialogue

3.3.6.10 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.

3.3.6.11 HALF SPEED?

Enter YES if the local modem is to operate at half speed. Enter NO if it is to operate at full speed. (Enter NO for the Hardwired Serial Interface.) The dialogue skips to Step 3.3.6.13.

3.3.6.12 SPEED CHANGEABLE?

Enter YES if the speed of the line is changeable. Enter NO if the line speed is fixed. (Enter NO for the Hardwired Serial Interface.)

3.3.6.13 TRANSMISSION SPEED?

Enter the transmission speed of the line in characters per second. For the Hardwired Serial Interface, enter 250000.

3.3.6.14 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 (and for the Hardwired Serial Interface)
1 = Half duplex

3.3.6.15 PREFERRED BUFFER SIZE?

If no other subsystems are using the line, enter 4096. Where there are other subsystems, enter the buffer size (in words) most suitable to the device being configured, but no more than 4096.

3.3.6.16 DRIVER CHANGEABLE?

Enter NO.

3.3.6.17 DRIVER OPTIONS?

Enter 0.

3.3.6.18 DRIVER NAME?

Enter CSSBSCO for SSLC.
Enter CSHBSCØ for Hardwired Serial Interface.

3.3.6.19 CONTROL LENGTH?

Enter 0.

Step No. **Dialogue**

3.3.6.20 PHONELIST?

Enter YES to provide a default phone number list. Enter NO if none provided. A NO response causes the next step to be skipped.

NOTE

The Emulator will sequentially step through the default phone number list if there is no specified number. The default phone number list is overridden at run time by specifying a phone number for the ;CONNECT=parameter in the #RJLINE command.

3.3.6.21 PHONE NUMBER?

Enter a string of numbers and hyphens, but not more than 20 characters. This can be repeated until a carriage return is entered.

3.3.6.22 LOCAL ID SEQUENCE?

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.

3.3.6.23 REMOTE ID SEQUENCE?

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.

3.3.6.24 DEVICE CLASSES?

Enter one or more device class names, separating the names with commas and terminating the list with a carriage-return. (For further information about device class names, see Step 3.3.16.) The dialogue returns to Step 3.3.

Step No.	Dialogue

3.3.7 RECORD WIDTH?

Enter the record width 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.

3.3.8 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.

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.

3.3.9 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.

3.3.10 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.

Step No.	Dialogue

3.3.11 INTERACTIVE?

To specify that this is an interactive device (I), enter YES.

Otherwise, enter NO.

Note: Disc devices should *not* be interactive.

3.3.12 DUPLICATIVE?

To specify that this is a duplicative device (D), enter YES.

Otherwise, enter NO.

Note: Disc devices should *not* be duplicative.

3.3.13 INITIALLY SPOOLED?

To designate this device as being spooled at cold-load (S), enter YES.

Otherwise, enter NO.

3.3.14 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.

3.3.15 DRIVER NAME?

Enter the name of the program file containing the driver for this device (Appendix A). 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.)

3.3.16 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.

When the input is complete, enter a carriage return to return to Step 3.3.

Step No.	Dialogue
3.4	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.

NOTE

Each concurrent batch job to be executed requires 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.

3.5 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 3.1.

To suppress the list, enter NO.

NOTE

The prompt in Step 3.5.1, below, appears only if you have configured an SSLC device into the system.

3.5.1 LIST CS DEVICES?

Enter YES to list the characteristics of the new CS device configuration. Enter NO to suppress the listing.

3.6 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 avoid class changes and skip to Step 3.6.5, enter NO.

Step No.	Dialogue

3.6.1 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 8-31)
OUT - serial output (device types 16-39)
DA - Direct access (device types 0-7)
I/O,C - Input/output, concurrent devices (device types 16-23)
I/O,NC - Input/output, non-concurrent devices (device types 16-31)

LOGICAL DEVICES are the logical device numbers of all devices specified for this class in the I/O configuration.

3.6.2 DELETE CLASSES?

To delete previously defined classes, enter YES.

Otherwise, enter NO to skip to Step 3.6.3.

3.6.2.1 CLASSES?

Enter names, separated by commas, of classes to be deleted.

3.6.3 ADD CLASSES?

To define new classes or to add devices to previously defined classes, enter YES. To skip to Step 3.6.4, enter NO.

3.6.3.1 CLASS NAME?

To define a new class or to add devices to a previously defined class, enter class name.

To skip to Step 3.6.4, enter a carriage return.

3.6.3.2 LOGICAL DEVICE #S?

Enter logical device numbers, separated by commas, in the preferred order of allocation.

If it is not necessary to ask question 3.6.3.3, then a return is made to Step 3.6.3.1.

3.6.3.3 (IN, OUT, OR IN/OUT?) (CONCURRENT OR NON-CONCURRENT?)

Depending upon the types of the devices within the class, these questions may be asked to determine the desired device class access type. Second question is asked only if the response to the first is IN/OUT.

Enter: IN
OUT
IN/OUT,NC
IN/OUT,C

Return to STEP 3.6.3.1.

3.6.4 LIST CLASSES?

To list the device classes and logical devices contained therein, enter YES.

To suppress the listing, enter NO.

3.6.5 LIST I/O DEVICES?

To print a listing of the new I/O configuration, enter YES.

To suppress the listing, enter NO.

Step No.	Dialogue
	NOTE
	The prompts in Steps 3.6.5.1 through 3.6.5.5, below, appear only if an SSLC is configured or if additional drives exist (for the SSLC driver-changeable option). If neither case exists, the dialogue skips to Step 4.
3.6.5.1	ADDITIONAL DRIVER CHANGES?
	To prepare for additional driver changes, enter YES.
	To skip to Step 4, enter NO.
	NOTE
	The prompts in Steps 3.6.5.2 through 3.6.5.3.1, below, appear only if additional drivers are already configured; otherwise, the dialogue skips to Step 3.6.5.4.
3.6.5.2	LIST ADDITIONAL DRIVERS?
	To print a listing showing the presently-configured additional drivers, enter YES.
	To suppress this listing, enter NO.
3.6.5.3	DELETE DRIVERS?
	To delete an existing additional driver, enter YES.
	To skip to Step 3.6.5.4, enter NO.
3.6.5.3.1	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.

Step No.	Dialogue
3.6.5.4	ADD DRIVERS? To prepare for adding drivers, enter YES. To skip to Step 3.6.5.5, enter NO.
3.6.5.4.1	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.
3.6.5.5	LIST ADDITIONAL DRIVERS? To print a listing showing the presently-configured additional drivers, enter YES. To suppress this listing, enter NO.

Step No.	Dialogue
4	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 5, enter NO.
4.1	SHAREABLE AREA 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.
4.2	PROGRAM AREA 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.
4.3	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.
4.4	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.
4.5	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.
4.6	TERMINAL BUFFERS = <XXX>? To change the number of terminal buffers in the system from XXX, enter the new value. To retain the current value, enter a carriage return.

Step No.	Dialogue
4.7	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.
4.8	MEMORY MANAGEMENT TABLE = <XXXX>? To change the size of the table from XXXX to another value, enter the new value. To retain the current value, enter a carriage return.
4.9	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.
4.10	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.
4.11	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.
4.12	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.
5	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.

Step No. **Dialogue**

- 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 6, enter NO.

5.1 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).

5.2 DELETE GOBAL 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 5.3, enter NO.

5.2.1 ENTER RIN NUMBER?

To delete a currently-assigned global RIN, enter the RIN number.

This step is repeated until a cariage return is entered.

NOTE

Since global RIN's are premanently 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.	Dialogue
5.2.2	LIST GLOBAL RINS? To list the updated global RIN's (as in Step 5.1), enter YES. To suppress the listing, enter NO.
5.3	# 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.
5.4	# 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.
5.5	# 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.
5.6	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.
5.7	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.
5.8	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.	Dialogue
5.9	LIST MESSAGE CATALOG? To list the current message catalog, enter YES. To suppress the listing, enter NO.
5.10	MESSAGE CATALOG CHANGES? To create a new message catalog from supplied file, enter YES. To skip to Step 6, enter a carriage return.
5.10.1	MESSAGE CATALOG INPUT FILE NAME? Name of disc file or formal designator to define the input file from which new catalog is to be built.
5.10.2	LIST MESSAGE CATALOG? To list new catalog, enter YES. To suppress the listing, enter NO.
6	LOGGING CHANGES? To prepare for changes to the logging characteristics of the system, enter YES. To bypass such changes and proceed to Step 7, enter NO.
6.1	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.
6.2	CHANGE STATUS? To prepare for changes to the logging status, enter YES. If no changes are desired, enter NO to skip to Step 6.3.
6.2.1	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.

Step No. **Dialogue**

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	Spooling log record
9	Line disconnection
10	Line close
11	I/O error

NOTE

Event 1 must be ON for any logging to take place. If event 2 is on, the default input priority for jobs and sessions is 8; if event 8 is on, this default output priority is 8. Otherwise, the normal default is 13 for both input and output priorities.

Step 6.2.1 is repeated until a carriage return is entered.

6.2.2 LIST LOGGING STATUS?

To list the updated logging status, respond with YES. To suppress the listing, enter NO.

6.3 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.

6.4 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.

Step No.	Dialogue
7	DISC ALLOCATION CHANGES? To prepare for disc allocation changes, enter YES. To bypass such changes and proceed to Step 8, enter NO.
7.1	VIRTUAL MEMORY = <XXXXX>.? (RELOAD option only.) To change the size of the area on disc used for virtual memory from XXXXX sectors to another value, enter the new value. To retain the current value, enter a carriage return.
7.2	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 6000 sectors. To retain the present maximum size, enter a carriage return.
7.3	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* — *LOG DEV #* *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.
7.4	DELETE VOLUME? (RELOAD option only.) To prepare to delete a volume, enter YES. To bypass deletion and skip to Step 7.5, enter NO.

Step No.	Dialogue
7.4.1	ENTER VOLUME NAME? (RELOAD option only.).
	To delete a volume, enter the volume name. (When the name is entered, the question is repeated.)
	Otherwise, enter a carriage return.
7.5	ADD VOLUME?
	To prepare to add a volume, enter YES.
	To bypass addition and skip to Step 7.6, enter NO.
7.5.1	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.
7.6	LIST VOLUME TABLE?
	To list the disc volumes and their currently assigned logical device numbers (as in Step 7.3), enter YES. In this listing, volumes just added (in Step 7.5) will have logical device numbers of zero.
	To suppress this listing, enter NO.
7.7	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.
7.8	# OF SECTORS PER SPOOLFILE EXTENT?
	Enter the size, in sectors, for each spoolfile extent. This must be a value between 128 and 1024.
8	SCHEDULING CHANGES?
	To prepare for changes to the scheduling queue, enter YES.
	To bypass these changes and proceed to Step 9, enter NO.
8.1	TIME QUANTUM = <XXXXX>.?
	To change the time quantum in milliseconds, enter the new value. To keep the old value, enter a carriage return.

Step No.	Dialogue
8.2	TERMINAL PRIORITY = <XXX> ? To change the priority assigned to a time-shared process when it completes a terminal read operation, enter the new value. To keep the old value, enter a carriage return.
8.3	CS PRIORITY LIMIT = <XXX> ? To change to the lowest priority to which a process in the CS subclass can fall, enter the new value. To retain the old value, enter a carriage return.
8.4	DS PRIORITY LIMIT = <XXX> 0 To change to the lowest priority to which a process in the DS subclass can fall, enter the new value. To retain the old value, enter a carriage return.
9	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 10, enter NO.
9.1	MAX # OF CONCURRENT RUNNING PROGRAMS = <XXX>.? To change the maximum number of concurrent running programs from XXX, enter the new value. To retain the current value, enter a carriage return.
9.2	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.
9.3	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.
9.4	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.

Step No.	Dialogue
9.5	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.
9.6	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.
9.7	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.
10	SYSTEM PROGRAM CHANGES? To prepare to replace a program belonging to the system, enter YES. To proceed directly to Step 11, enter NO.
10.1	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.
11	SYSTEM SL CHANGES? To prepare for changes to the System Library (SL.PUB.SYS) enter YES. Otherwise, enter NO to skip to Step 12.
11.1	LIST LIBRARY? To list the names of the code segments in the System Library and their entry-points and external procedures, enter YES. To suppress this listing, enter NO.

Step No.	Dialogue

11.2 DELETE SEGMENT?

To prepare for deleting a code segment from the System Library (SL.PUB.SYS), enter YES.

To proceed directly to Step 11.3, enter NO.

11.2.1 ENTER SEGMENT NAME?

To delete a code segment from the System Library, enter the name of that segment. (When the segment name is entered, the question is repeated.)

Otherwise, enter a carriage return.

11.3 REPLACE SEGMENT?

To prepare for replacing a code segment in the System Library, enter YES.

To proceed directly to Step 11.4, enter NO.

11.3.1 ENTER SEGMENT NAME,USLFILE NAME [,S/C/P]?

To replace a code segment in the System 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:

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 USLLFILE name which is in error (typographic input error), an error message results and the Configurator proceeds back to Step 11.2. In this regard the segment which was to have been replaced has, in fact, been *deleted*. You may proceed to Step 11.4 and attempt to add the USL file.

Step No.	Dialogue

11.4 ADD SEGMENT?

To prepare for adding a code segment to the System Library, enter YES.

Otherwise, enter NO to skip to Step 11.5.

11.4.1 ENTER SEGMENT NAME, USLFILE NAME [,S/C/P]?

To add a code segment to the System 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:

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 11.2.

11.5 LIST LIBRARY?

To list the updated System Library, enter YES.

To suppress this listing, enter NO.

12 ENTER DUMP DATE?

To copy only the modified operating system to tape, enter a carriage return; the dialogue skips to Step 13.

To copy the MPE system, the current accounting structure, and all files to tape, 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.

Step No. **Dialogue**

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.

12.1 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 in 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.

13 The operator is now requested to assign the magnetic tape device on which you have arranged for a fresh magnetic tape to be mounted. After operator assignment, the system is copied to tape (multi-reel files). It then can be loaded and initialized as directed under the heading *System Start-Up and Modification.*

Any file belonging to the system that was not copied for some reason will be noted. If that file was to replace a system program, the program name will follow in parentheses.

If a response other than a carriage return was entered in answer to the ENTER DUMP DATE? question in Step 12, the list and count of files will be provided as described in Step 12.1.

To denote termination of the Configurator/User Dialogue, the following message is printed:

END OF SUBSYSTEM

EXAMPLE OF SYSTEM BACK-UP ONLY

The Configurator/user dialogue 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 magnetic tape. Prior to entering the dialogue, the user logs on to the system and inputs at least a file reference to a magnetic tape. 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.	Dialogue
0	ANY CHANGES? NO Reply NO (or carriage return) to omit changing the system and to skip to Step 12.
12	ENTER DUMP DATE? Enter carriage return. This is the null case of the Configurator. The operator is now requested to assign the magnetic tape device on which you have arranged for a fresh magnetic tape to be mounted. After operator assignment, the system (only) is copied to tape. In this regard, note that the account, group, and user structure is *not* copied to the tape. However, the system account (SYS) is copied.
13	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 dialogue proceeds with the Configurator (:SYS DUMP) 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 dialogue, the user logs onto the system and inputs at least a file reference to a magnetic tape.

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 line printer respectively.

In this example to save the system and all files, the dialogue is as follows:

Step No.	Dialogue
0	ANY CHANGES ? NO Enter NO (or carriage return) to omit modification to the system and to skip to Step 12.
12	ENTER DUMP DATE? 0 Enter a zero. 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 usually will be a multiple magnetic tape function if a great many files exist on the system. 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 tape. This defeats the purpose of a full system and file backup. Since we are saving files along with the system, the Configurator proceeds to Step 12.1.
12.1	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 magnetic tape device on which you have arranged for a fresh magnetic tape to be mounted. After operator assignment, the system, all accounting structures, and all files on the system are copied to tape. If multiple reels of magnetic tape are required, the operator will be requested to mount the succeeding reels at the appropriate time.
13	To denote termination of the Configuration program, the following message is printed: 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 date placed in the file label at the time of modification (during the file close processing). 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.

In the preceding example, everything was saved on a magnetic tape or tapes. 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 the various files that have been modified since the full system and file back-up.

In the next example, let us assume that on Friday, June 18, 1976, you did a full system save as in the previous example. On Monday, June 21, the users on the system created many new files. When the day is done and there are no other users on the system, you as the System Supervisor log onto the system and enter at least a file reference to a magnetic tape. A typical example is:

```
:FILE SAVSOME; DEV=TAPE
:FILE LISTSOME; DEV=LP
:SYSDUMP *SAVSOME, *LISTSOME
```

(Assume that device classes TAPE and LP are a magnetic tape and a high speed line printer respectively.)

The dialogue is as follows:

Step No.	**Dialogue**
0	ANY CHANGES? NO Enter NO or carriage return to omit modifications to the system and to skip to Step 12.
12	ENTER DUMP DATE? 6/18/76 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 12. All files whose modification date is on or after than the one which you have input will be saved on the magnetic tape. 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 second tape and when you are asked if there are more files, you mount the SYSDUMP tapes from 6/18/76 until there are no more.
12.1	LIST FILES DUMPED? To obtain a listing of all the files which have been modified since the date input in Step 12, enter YES. To suppress the listing, enter NO. A list showing 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. To obtain a listing of all the files which have been modified on or after the date have arranged for a fresh magnetic tape to be mounted. After operator assignment, the system, all accounting structures, and all files modified since the data entered in Step 12 will be copied to magnetic tape.
13	To denote termination of the Configurator, the following message is printed: END OF SUBSYSTEM

ACCOUNTING STRUCTURE

In the previous discussion of the Configurator, it has been stated that your account, group, user and directory information is kept on the magnetic tape (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 stream 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 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, since he need not 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 provides an option of five types of cold-load.

- WARMSTART cold-loads the system from the system disc. This is the standard operating procedure for restarting the system, since it is the only option which results in the recovery of incompletely processed, spooled jobs and spoolfiles. On any other option, any spoolfiles found in the system are deleted.

  If a COOLSTART, COLDSTART, UPDATE, or RELOAD is aborted for any reason, the next cold load may not be a WARMSTART. A violation of this results in an error message and halt.

- COOLSTART cold-loads the MPE System from the system disc. (This is the standard operating procedure when a system is routinely shut down at night and brought up again the next day.) All resident 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 environment present prior to the last shutdown is not retained. Thus, all jobs or sessions in progress at shutdown are lost.

- COLDSTART cold-loads the MPE System from magnetic tape, using the system files and input/output device configuration on that tape while retaining the user files, directory and accounting information, and assigned resource identification numbers (global RIN's) currently on disc. This allows modification of the system configuration while retaining the user's information. COLDSTART is commonly used to allow an installation to keep several cold-load tapes, each with a different configuration. If aborted, a COLDSTART must be followed by a COLDSTART, UPDATE, or RELOAD. Any violation of this results in an error message and a halt.

- UPDATE cold-loads the MPE System from magnetic tape, using the system files from that tape; the input/output devices, system configuration, directory, accounting information, and global RIN's from the system disc; and the user files from disc. This is the standard

operating procedure used 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 is aborted, it must be followed by a COLDSTART, UPDATE, or RELOAD. Any violation of this results in an error message and a halt.

- RELOAD cold-loads the entire MPE System, including all system files and system and input/output configuration information, from magnetic tape. This option assumes almost no information on the disc. (This is discussed further under the heading *Disc Organization.*) If any user files were dumped on the tape, the directory, accounting information, assigned global RIN's, and user files are restored to the disc from the tape. If no user files were dumped, a directory is created with the SYS account, PUB group, and MANAGER user. RELOAD is normally done when installing the system from the first MPE tape received from Hewlett-Packard, or when restoring the system following a disc crash (from a tape generated by the user through a :SYSDUMP command). If a RELOAD is aborted for any reason, the next cold load must also be a RELOAD.

  When reloading from multiple sets of tapes created by the :SYSDUMP command, the first reel of the latest tape set should be used for cold-loading; it contains the up-to-date directory and accounting information. If not all of the files on the system are contained in this tape set, an additional set will be requested.

  No start-up option permits resumption of user batch jobs or interactive sessions interrupted by a system shut-down or hardware or software failure; such jobs must be re-initiated from the beginning by the user.

  If a RELOAD is aborted for some reason, the next cold load also must be a RELOAD. If a COLDSTART or UPDATE is aborted, the next cold load must be a COLDSTART, UPDATE, or RELOAD. Any violations of this rule result in an error message and a halt.

  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.

  When the system is started, the initialization of blank 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.

  When using the Initiator to modify MPE, the System Supervisor requires certain background information. The information he needs depends on the type of change he wishes to make, as noted below.

  Table 6-1 shows the origin of MPE system components for the various cold-load options.

CHANGING MAIN MEMORY SIZE

MPE runs on HP 3000 Computer Systems with main-memory of the following sizes (all sizes are in K-words where K=1024):

64K	160K	256K
96K	192K	
128K	224K	

The size specified by the user during configuration should be the actual size of main-memory delivered with the machine on which this MPE configuration is to run.

CHANGING I/O DEVICE CONFIGURATION

All input/output devices can be changed in exactly the same way as for :SYSDUMP. See Steps 3 through 3.6.5 of the :SYSDUMP dialogue for details

Table 6-1. Source of MPE System Components

MPE COMPONENT	WARMSTART	COOLSTART	UPDATE	COLDSTART	RELOAD*
MPE Programs, System Library	disc	disc	tape	tape	tape
I/O & System Configurations	disc	disc	disc	tape	tape
Accounting Info, File Directory, Volume Table & User Files	disc	disc	disc	disc	tape
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 given are only approximations. 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), Disc Free Space Table and user files.

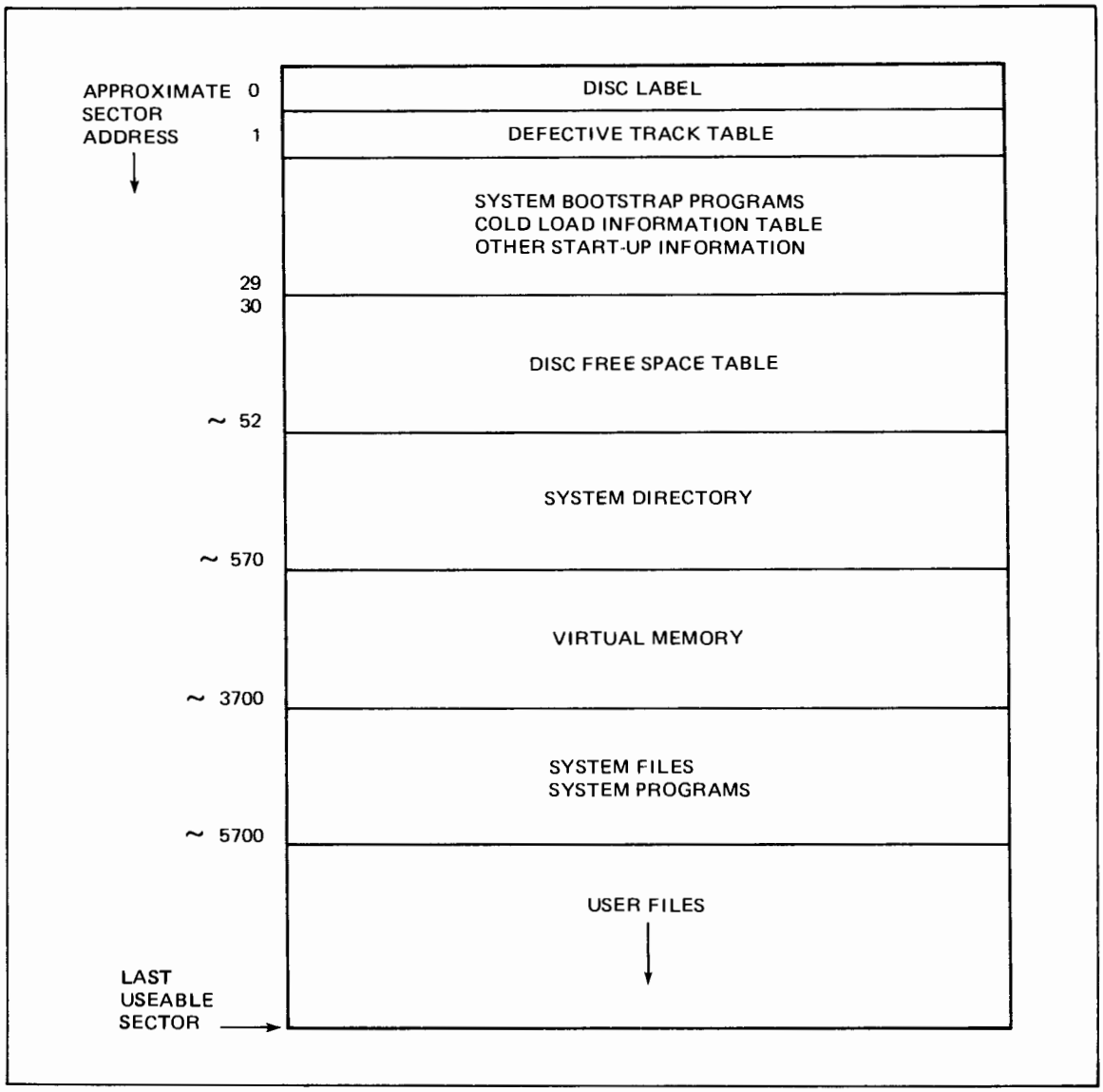


Figure 6-1. Typical System Disc Layout

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 with the disc pack installed. Each platter of the disc pack has two surfaces used for storing information (except the top and bottom surfaces). A cylinder is the area of all surfaces of the pack where all 20 heads can access a particular area of a platter surface. The head mechanism moves the entire group of heads from cylinder 0 through cylinder 405, depending on what cylinder has been requested by the software. A track is the area under one head on one surface. Therefore, in this case, we could access 20 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 or WARMSTART, 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

A common misconception is that the RELOAD option builds the entire system from the magnetic tape to the discs. Actually, what happens is as follows: When the RELOAD is performed, checks are made for the validity of the VOLUME label on disc. If the format of the name is a valid ASCII string (and certain other validity checks are passed), then the Initiator will use the disc as a system or user disc and will not over-write certain table information on the disc.

The Initiator automatically will assign sector 1 of track 0 as the *defective tracks 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).

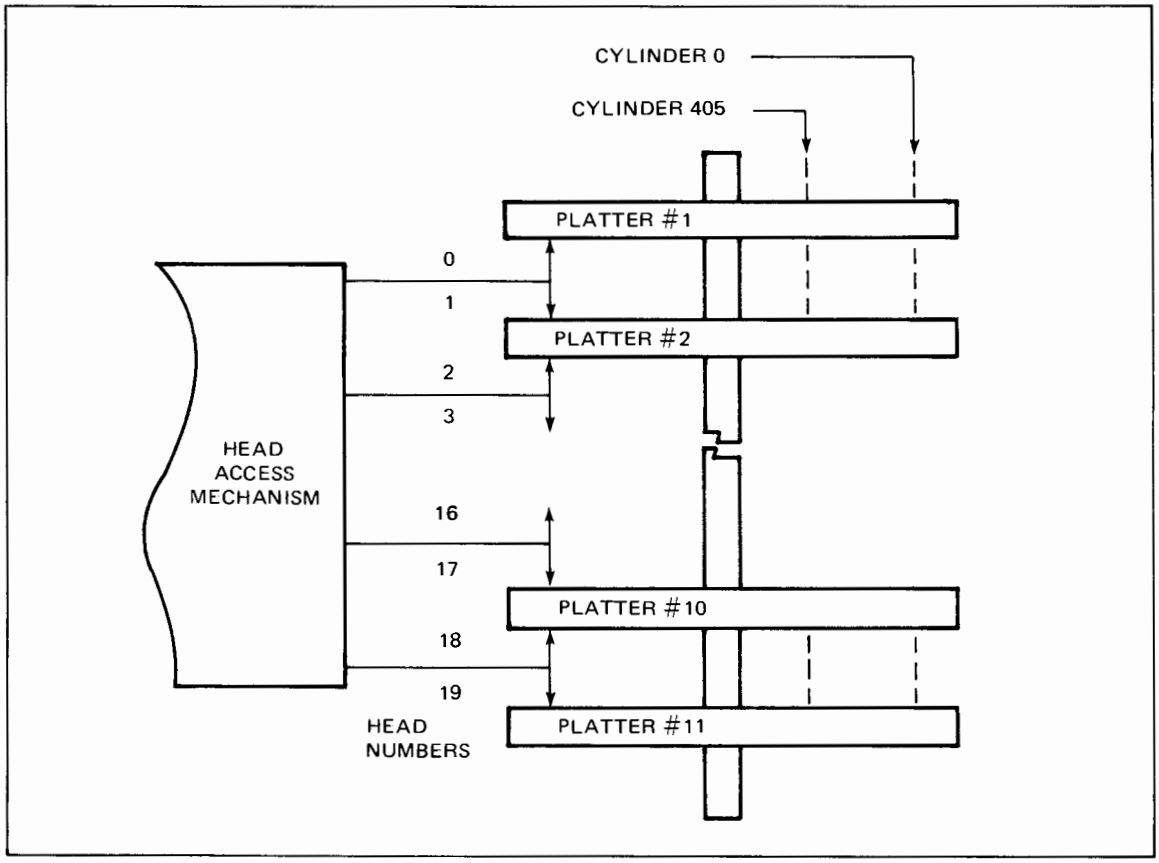


Figure 6-2. Disc Drive Structure

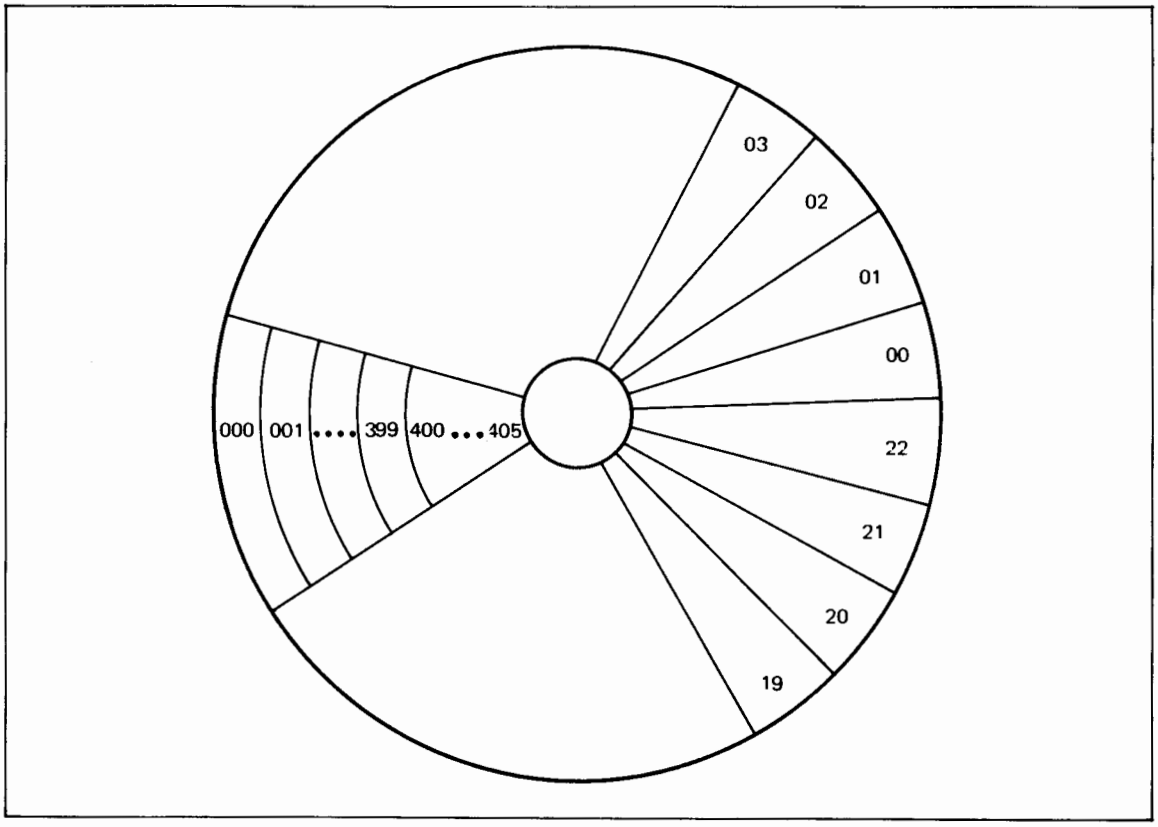


Figure 6-3. Disc Platter Format

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 the moving-head type, 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 moving-head 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. An error occurs such that track 701 cannot be read by the system. An entry then is made in the defective tracks table for that disc pack. The user attempting to read the file receives an error message an eventually gives up trying to read the file. When the operator/System Supervisor next cold-loads the system, the defective track entry will be 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 table. The entry is made permanent in the table and cannot be removed. The Initiator then makes a change to the disc free space table 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.

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.

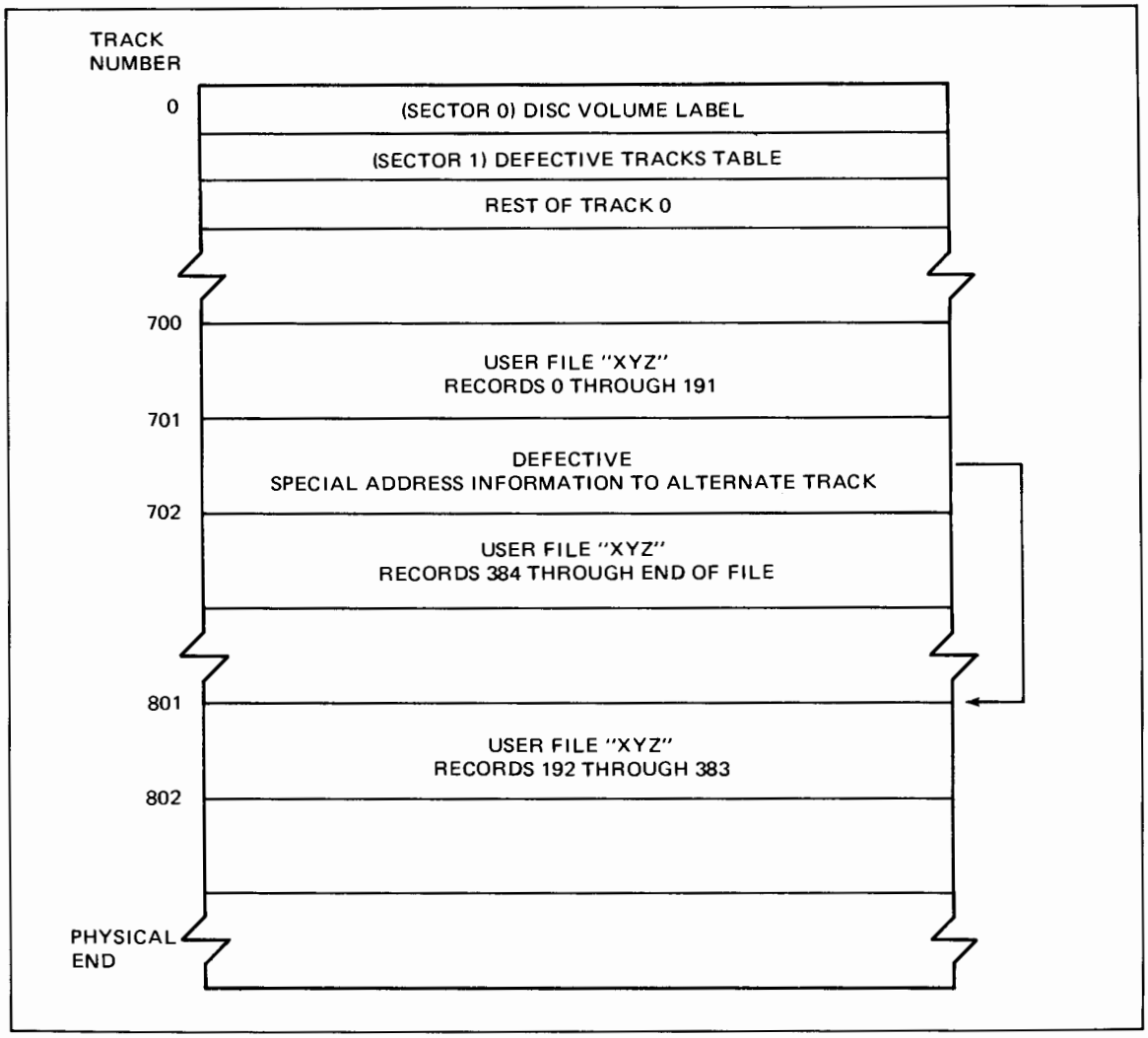


Figure 6-4. Disc With a Track Reassignment

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 virtual memory area may not be REASSIGNed except during a RELOAD, in which case the entire virtual memory area is moved to a place on the disc large enough for contiguous access to all of virtual memory.
- Tracks in the alternate track area of a moving-head disc pack may *not* be REASSIGNed. The reason for disallowing this is that the hardware controller 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, 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 above disc error and recovery function, it can be seen that a regular method of saving files to magnetic tape is extremely important.

INITIATOR/USER DIALOGUE

The Initiator/operator or System Supervisor dialogue format (Initiator output and the user's response to it) is exactly the same as previously discussed for the :SYSDUMP command.

To begin the dialogue, follow these steps:

a. Mount the magnetic tape containing the system and any associated new files and configuration data on the magnetic tape device. The tape should be mounted on the tape unit specified by DRT entry number 6, unit 0.

b. Set the SWITCH REGISTER on the computer control panel with the "READ" setting and DRT number of the device. For magnetic tape the setting is octal 003006. For a disc device the setting is octal 000*nnn* where *nnn* is the DRT number of the system disc in octal.

c. While pressing (and holding) the ENABLE switch, press the LOAD switch. This action causes the cold-load operation to begin. The Initiator program will be read into main memory.

d. Press the RUN switch. The initiator program is now in execution. Next, press the carriage return key 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 dialogue with the user by printing the information shown in table 6-2 on the console. (In the dialogue shown in table 6-2, output from the Initiator is shown verbatim in uppercase letters, and input from the user is described in mixed upper and lower-case letters.)

   If cold-loading is done from magnetic tape, the MPE Initiator begins its dialogue at step 1. If cold-loading is done from disc, the dialogue begins with step 0.

   For the purpose of this dialogue, you should be familiar with the discussion and dialogue of the :SYSDUMP command. The dialogue in table 6-2 is similar to that for :SYSDUMP and will only be explained in detail where the two operations differ.

Table 6-2. Initiator-User Dialogue

STEP NO.	DIALOGUE
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 dialogue is skipped (i.e., Steps 3 through 4 and Steps 5.1 through 10). Whether WARMSTART or COOLSTART is selected, skip to Step 2.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
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 tape, the Initiator proceeds to Step 1.1 (below). If no user files exist on the tape, the Initiator skips to step 1.2. If either COLDSTART or UPDATE is selected, the Initiator skips to Step 2.
1.1	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 2. 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. **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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	ACCOUNTS This option loads the system, creates a directory from the magnetic tape, 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 was written by the :SYSDUMP Configurator program. This option is useful because files saved on magnetic tape 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 1.2) 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.
1.2	NO USER FILES ON TAPE — DO YOU WANT TO RELOAD? This implies that when the system tape 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 1, enter NO.
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 core dump is taken for the analysis. To suppress this map, enter NO. If this a WARMSTART, control skips to Step 5.
3	ANY CHANGES? To proceed with changes to the current configuration, enter YES. To maintain the current configuration, enter NO. If this is a RELOAD without changes, the Initiator skips to Step 5.8. Otherwise, it skips to Step 5 for COLDSTART and UPDATE options.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
3.1	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, 226, 256.
3.2	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 5, enter NO.
3.2.1	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 3.1 of the :SYSDUMP dialogue for the format of the listing. NOTE The prompt in Step 3.2.1.1, below, appears only if a synchronous single-line controller (SSLC) was previously configured into the system.
3.2.1.1	LIST CS DEVICES To print a list of the characteristics of all SSLCs 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 3.1.1 of the :SYSDUMP dialogue for the format of the listing.
3.2.2	HIGHEST DRT=<XX>? In the output, 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 deteriming this value.) To change XX, enter the new value desired. To maintain the current XX, enter a carriage return.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
3.2.3	LOGICAL DEVICE #? To specify a device to be added or removed, enter the logical device number of that device. To skip to Step 3.2.4, enter zero or a carriage return.
3.2.3.1	DRT #? To add a device, enter its DRT entry number. To remove a device and return to Step 3.2.3, enter zero.
3.2.3.2	UNIT #? Enter the physical hardware unit number of the device, if the device shares its controller with other devices. Otherwise, enter zero to continue.
3.2.3.2.1	CHANNEL # If the device is to be on a multicontroller channel, enter channel number; if not, enter zero.
3.2.3.3	TYPE? Enter the device type (Appendix A).
3.2.3.4	SUB-TYPE? Enter the device sub-type in the range 0 to 15 (Appendix A).
3.2.3.5	TERM TYPE? This question is asked only if type is 16. To specify default terminal type to be used at logon, enter a value from the list below. **Terminal** — **Term Type** ASR-33 — 0 ASR-37 — 1 ASR-37 — 2 Execuport 300 — 3 HP 2600A or Datapoint 3300 — 4 Memorex 1240 — 5 GE Terminet 300 or 1200 — 6 HP 2615A — 9 HP 2640A or HP 2644 (Character mode) — 10 HP 2640A or HP 2644 (block/line mode) — 11 NOTE If term type is not known, enter carriage return.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	NOTE The prompts in Steps 3.2.3.5.1 through 3.2.3.5.24, below, appear only if you are configuring an SSLC. If you are *not* configuring an SSLC, the dialogue skips to Step 3.2.3.6.
3.2.3.5.1	PROTOCOL? To define the SSLC protocol, enter 1.
3.2.3.5.2	LOCAL MODE? 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 HP 2780/3780 Emulator is usually a primary station.
3.2.3.5.3	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
3.2.3.5.4	RECEIVE TIMEOUT? Enter the positive number of seconds the Emulator will wait to receive text before terminating the read mode. Entering a carriage return provides a 20-second timeout. NOTE For all timeout responses: 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.
3.2.3.5.5	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.
3.2.3.5.6	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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	NOTE Steps 3.2.3.5.7 through 3.2.3.5.9 apply only to SSLCs with switched lines *not* connected via modems. If the SSLC is connected through a modem, the dialogue skips to Step 3.2.3.5.10. If the SSLC is hardwired, the dialogue skips to Step 3.2.3.5.12.
3.2.3.5.7	DIAL FACILITY? Enter YES when calls can be dialed from the local station. Enter NO when they cannot.
3.2.3.5.8	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.
3.2.3.5.9	AUTOMATIC ANSWER? Enter YES if the local modem can automatically answer calls. Enter NO if manual answering is required.
3.2.3.5.10	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.
3.2.3.5.11	HALF SPEED? Enter YES if the local modem is to operate at half speed. Enter NO if it is to operate at full speed. The dialogue skips to Step 3.2.3.5.13.
3.2.3.5.12	SPEED CHANGEABLE? Enter YES if the speed of the line is changeable. Enter NO if the line speed is fixed.
3.2.3.5.13	TRANSMISSION SPEED? Enter the transmission speed of the line in characters per second.
3.2.3.5.14	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 1 = Half duplex
3.2.3.5.15	PREFERRED BUFFER SIZE? If no other subsystems are using the line, enter 4096. Where there are other subsystems, enter the buffer size (in words) most suitable to the device being configured, but no more than 4096.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
3.2.3.5.16	DRIVER CHANGEABLE? Enter NO.
3.2.3.5.17	DRIVER OPTIONS? Enter 0.
3.2.3.5.18	DRIVER NAME? Enter CSSBSC0.
3.2.3.5.19	CONTROL LENGTH? Enter 0.
3.2.3.5.20	PHONELIST? Enter YES to provide a default phone number list. Enter NO if none provided. A NO response causes the next step to be skipped. NOTE The Emulator will sequentially step through the default phone number list if there is no specified number. The default phone number list is overridden at run time by specifying a phone number for the ;CONNECT= parameter in the #RJLINE command.
3.2.3.5.21	PHONE NUMBER? Enter a string of numbers and hyphens, but not more than 20 characters. This can be repeated until a carriage return is entered.
3.2.3.5.22	LOCAL ID SEQUENCE? 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.
3.2.3.5.23	REMOTE ID SEQUENCE? 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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
3.2.3.5.24	DEVICE CLASSES? Enter one or more device class names, separating the names with commas and terminating the list with a carriage-return. (For further information about device class names, see Step 3.3.16 in the :SYSDUMP dialogue.) The dialogue returns to Step 3.2.3.
3.2.3.6	RECORD WIDTH? Enter the default physical record width for the device. (See Step 3.3.7 of the :SYSDUMP dialogue and Appendix A.)
3.2.3.7	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 3.3.8 of :SYSDUMP dialogue for more details.) If this is not a job/session input device, enter zero.
3.2.3.8	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.
3.2.3.9	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.
3.2.3.10	INTERACTIVE? To specify that this is an interactive device, enter YES. Otherwise, enter NO or a carriage return.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
3.2.3.11	DUPLICATIVE? To specify that this is a duplicative device, enter YES. Otherwise, enter NO or a carriage return.
3.2.3.12	INITIALLY SPOOLED? To designate this device as being spooled at cold-load, enter YES. Otherwise, enter NO or a carriage return.
3.2.3.12.1	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.
3.2.3.13	DRIVER NAME? Enter the name of the program file containing the driver for this device. User program driver names as well as those in Appendix A may be used. If driver name is preceded by an asterisk, the driver will be main memory resident at all times.
3.2.3.14	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. When the input is complete, enter a carriage return to return to Step 3.2.3.
3.2.4	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. NOTE Each concurrent batch job to be executed requires 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.
3.2.5	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 3.1 of :SYSDUMP dialogue. To suppress the list, enter NO or a carriage return.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	NOTE The prompt in Step 3.2.5.1, below, appears only if you have configured an SSLC device into the system.
3.2.5.1	LIST CS DEVICES? Enter YES to list the characteristics of the new CS device configuration. Enter NO to suppress the listing.
3.2.6	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 Step 4, enter a carriage return.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
3.2.6.1	LIST CLASSES? To list the device classes and the logical devices contained therein, enter YES. See Step 3.6.1 of :SYSDUMP dialogue for the format of the list. To suppress the listing, enter NO or a carriage return.
3.2.6.2	DELETE CLASSES? To delete previously defined classes, enter YES. Otherwise, enter NO or a carriage return and skip to Step 3.2.6.3.
3.2.6.2.1	CLASSES? Enter the names, separated by commas, of classes to be deleted.
3.2.6.3	ADD CLASSES? To define new classes or to add devices to previously defined classes, enter YES. To skip to Step 3.2.6.4, enter NO or a carriage return.
3.2.6.3.1	CLASS NAME? To define a new class or to add devices to a previously defined class, enter class name. To skip to Step 3.2.6.4, enter a carriage return.
3.2.6.3.2	LOGICAL DEVICE #s? Enter logical device numbers, separated by commas, in the preferred order of allocation. If it is not necessary to ask question 3.2.6.3.3, then a return is made to Step 3.2.6.3.1.
3.2.6.3.3	[IN,OUT, OR IN/OUT?] [CONCURRENT OR NON-CONCURRENT?] Depending upon the types of the devices within the class, these questions may be asked to determine the desired device class access type. Enter: IN — IN/OUT, NC OUT — IN/OUT, C Return to Step 3.2.6.3.1
3.2.6.4	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.
3.2.6.5	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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	NOTE The prompts in Steps 3.2.6.5.1 through 3.2.6.5.5, below, appear only if an SSLC is configured or if additional drives exist (for the SSLC driver-changeable option). If neither case exists, the dialogue skips to Step 4.
3.2.6.5.1	ADDITIONAL DRIVER CHANGES? To prepare for additional driver changes, enter YES. To skip to Step 4, enter NO. NOTE The prompts in Steps 3.2.6.5.2 through 3.2.6.5.3.1, below, appear only if additional drivers are already configured; otherwise, the dialogue skips to Step 3.2.6.5.4.
3.2.6.5.2	LIST ADDITIONAL DRIVERS? To print a listing showing the presently-configured additional drivers, enter YES. To suppress this listing, enter NO.
3.2.6.5.3	DELETE DRIVERS? To delete an existing driver, enter YES. To skip to Step 3.2.6.5.4, enter NO.
3.2.6.5.3.1	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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
3.2.6.5.4	ADD DRIVERS? To prepare for adding drivers, enter YES. To skip to Step 3.2.6.5.5, enter NO.
3.2.6.5.4.1	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.
3.2.6.5.5	LIST ADDITIONAL DRIVERS? To print a listing showing the presently-configured additional drivers, enter YES. To suppress this listing, enter NO.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
4	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 3.2.1. DEVICE OF DIFFERENT TYPE RANGES IN CLASS <CLASS> Device types (defined in :SYSDUMP dialogue Step 3.1) are divided into ranges for different kinds of devices: **Type Range** — **Class** 0-7 — Direct access. 8-15 — Serial input. 16-23 — Concurrent input/output. 24-31 — Non-concurrent input/output. 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. HIGHEST DRT ALLOWED IS <DRT> At least one device has been defined with a DRT number higher than the specified maximum. LOGICAL DEVICE <LDN> DOES NOT EXIST A device (LDN) specified as an output device is not defined in the configuration. MORE THAN ONE DEVICE FOR DRT <DRT> UNIT <UNIT> More than one logical device has been defined with the same DRT, unit, and subtype. 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 IN DRT 6 UNIT 0 A magnetic tape must be defined for DRT Entry Number 6, Unit 0.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
5	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 5.1. 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 3 the Initiator skips to Step 5.3. If all volumes were found and you had *not* responded YES to the ANY CHANGES? question in Step 3 the Initiator skips to Step 5.8. If this is a WARMSTART, control skips to Step 11.
5.1	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** **LOG DEV #** *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 5.4) 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 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
5.2	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 5), mount them, and cold load the system again or cold load using the RELOAD option and redefine the disc volume configuration (as discussed below).
5.3	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 5.5).
5.3.1	LIST VOLUME TABLE? To list the disc volume table (as in Step 5.1), 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 5.3.2 (if this is a RELOAD) or skips to Step 5.3.3 (if this is a COLDSTART, COOLSTART, or UPDATE).
5.3.2	DELETE VOLUME? To prepare to delete a volume, enter YES. To bypass deletion and skip to Step 5.3.3, enter NO or a carriage return.
5.3.2.1	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.
5.3.3	ADD VOLUME? To prepare to add a volume, enter YES. To bypass addition and skip to Step 5.3.4, enter NO or a carriage return.
5.3.3.1	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.
5.3.4	LIST VOLUME TABLE? To list the disc volumes and their currently-assigned logical device numbers (as in Step 5.1), enter YES. To suppress this listing, enter NO.
5.4	LIST DEFECTIVE TRACKS TABLE? To prepare for listing the defective tracks table which resides on any one of the disc volumes, enter YES. To bypass the listing and skip to Step 5.4.2, enter NO.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
5.4.1	LOGICAL DEVICE #? To list the defective tracks table 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 moving head 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 status — 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 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.
5.4.2	DELETE TRACK? To prepare for deleting tracks, enter YES. Otherwise, enter NO to skip to Step 5.5.
5.4.2.1	ENTER LDEV, CYLINDER AND HEAD? Note: Listing the defective tracks table of Step 5.4 will provide the input for this dialogue. 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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	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 5.5. 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 RESERVED AREA — CANNOT DELETE The track is in the area reserved for the disc label, bootstrap program, or disc free-space table, 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 DIRECTORY DELETE? The track is in the area previously used by the directory and this is a RELOAD using the COMPACT or RESTORE option. To delete the track (which will cause the directory to be moved) answer YES. Otherwise, answer NO.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	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 was printed, control returns to the beginning of this step.
5.5	If this system has been configured with the MPE Logging Facility enabled, the following question is printed: DISABLE LOGGING? To turn off logging until the next cold load, enter YES. Otherwise, respond with NO.
5.6	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.
5.7	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.
5.8	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 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 5.9.) (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 moving-head disc, this is followed by the message:

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE

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
2	150	200	203
3	300	400	406
4	300	400	411
5	300	400	411
6	300	400	411
7	120	120	125

(b) DEVICE <LDN> VOLUME <VOLNAME> NOT DEFINED IN TABLE
ENTER VOLUME NAME?

This means that the volume identified by VOLNAME, with the logical device number LDN, does ot 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 5.1), respond with YES. Otherwise, respond with NO. In either case, control then returns to Step 3.2.

(e) SUSPECT
SUSPECT ALT
UNREADABLE ALT } TRK LDEV#<LDN> CYL=<CYL> HEAD=<HEAD>

This message, applying to a moving-head 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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	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. 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 area between the volume label (disc label) and the end of the disc free space table. 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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	(k) **WARNING** IN RESERVED AREA RECOVER? A suspect track is located in the area between the disc label and the end of the disc free-space table. 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. (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, and this is not a RELOAD. Respond with YES to remove the entry or NO to leave it in the defective tracks table. (n) **WARNING** IN DIRECTORY DELETE, REASSIGN OR RECOVER? DELETE OR RECOVER? A suspect track is located in the area peviously assigned to the directory and this load is a RELOAD using the COMPACT, RESTORE, or ACCOUNTS options. The first form of the question is used for moving head discs and the second 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. (o) **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 from of the question will be used and responses DELETE, REASSIGN, or RECOVER are valid. If it is a RELOAD using the COMPACT, RESTORE, or ACCOUNTS options, the first form of the question will be used for moving head discs and the second form for fixed head discs. Responses are the same as in step n above. (p) DELETE OR RECOVER? The suspect track is located in no special area of a fixed head disc. Responses DELETE, RECOVER, or carriage return to ignore the problem, are valid. (q) 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.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
5.9	If any changes have been made to the defective tracks table in Step 5.8, the following question will be printed: LIST DEFECTIVE TRACKS TABLE? To prepare for listing the defective tracks table of any disc, respond with YES. Otherwise, respond with NO or a carriage return to skip to Step 5.10.
5.9.1	LOGICAL DEVICE #? Enter the logical device number of the disc for which the defective tracks table listing is desired, (as in Step 5.4.1). This question is repeated until a carriage return is entered signifying that no further listings are desired.
5.10	If any changes have been made to the volume table in Step 5.8, the following question is printed: LIST VOLUME TABLE? Respond with YES to obtain a listing of the volume table (as described in Step 5.1); otherwise, enter NO or a carriage return to bypass the listing.
6	If this load is not a RELOAD and the size of virtual memory on the disc differs from the configured size of virtual memory, the following message will be printed: **WARNING** VIRTUAL MEMORY SIZE ONLY CHANGED ON RELOAD This message will often occur when reconfiguring the system for a different main-memory size, since the default value for the size of the virtual memory differs for different main-memory sizes in the key records of the configuration data file known as CONFDATA. See :SYSDUMP discussion for more information.
7	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 This often occurs for the same reason described in Step 6.
8	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 *filename.groupname.accountname* . . . These files resided 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.
9	If this 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 10. The user files read from tape and written to disc can cause one of the following messages to be printed:

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
	(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. (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) NOT ALL FILES 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 dialogue 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) <XXXX> FILES NOT 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: NAME IS UNKNOWN message means the parity error occurred while reading the file label.)

Table 6-2. Initiator—User Dialogue (Continued)

STEP NO	DIALOGUE
	(i) DISC SPACE ERROR Configurator erred while attempting to return allocated disc space after detection of a parity error on tape. This halts the Configurator. Dialogue changes to Configurator-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.) The Initiator cannot recover from improper tape format, but a parity error in the trailer label will skip the first file of the next tape (prbably a continuation of the last file of the previous tape) and continue RELOAD with the next file. (k) IRRECOVERABLE PARITY ERRORS Initiator encountered parity errors in both the header and trailer labels of the tape and was not able to continue with the RELOAD operation. The Initiator aborts as a result.

Table 6-2. Initiator-User Dialogue (Continued)

STEP NO.	DIALOGUE
10	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 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.
11	DATE? Enter 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.
11.1	TIME? 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 of the hour.
11.2	(DATE AND TIME)? The date and time of day entered in Steps 11 and 11.1 are echoed for verification. If you respond with NO, the Initiator returns to Step 11. If the date and time are correct, enter YES or a 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 Dialogue 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.

NOTE

The Initiator will always write the load map to a disc file named LOAD.PUB.SYS on the system startups from *magnetic tape*, i.e. on a WARMSTART or COOLSTART it is not written to the disc. This disc file is a 32-byte ASCII file. It can be readily copied by FCOPY or used as an unnumbered text file by the EDITOR. It is created under the user name MANAGER.

COLD LOAD ERROR MESSAGES

One of the error messages shown in table 6-3 may appear while cold loading the system. Following the printing of the message, the machine halts.

```
01 ININ
02 FIRMWARESIM2 (1)
03 CIFILEM (5)
04 CILISTF (7)
05 CISUBS (13)
06 FILESYS7 (14)
07 FILESYS6A (15)
10 NRIO (20)
11 MESSAGE (21)
12 DATASEG (22)
13 PINT (27)
14 PCREATE (30)
15 CROUTINE (31)
16 EXCHANGEDB (41)
17 FILESYS6 (43)
20 FILESYS5 (46)
21 FILESYS4 (51)
22 FILESYS3 (55)
23 SPOOLING (61)
24 STKDUMP (62)
25 CXSTOREST (63)
26 FILESYS2 (64)
27 CIINIT (65)
30 MMDISKR (66)
31 FILESYS1A (67)
```

Figure 6-5. Load Map

```
 32 SEGUTIL (70)
 33 ABORTRAP (71)
 34 STORE (72)
 35 CHFCKER (73)
 36 UTILITY2 (74)
 37 UTILITY1 (76)
 40 CRIO (101)
 41 ALLOCATE (104)
 42 DIRC (106)
 43 SPOOLCOMS2 (110)
 44 ALLOCUTIL (111)
 45 MMCORER (112)
 46 CIORGMAN (113)
 47 DISKSPC (114)
 50 CIMISC (115)
 51 SPOOLCOMS1 (116)
 52 RINS (117)
 53 RESTORE (120)
 54 PROCMAIL (121)
 55 CIERR (123)
 56 MORGUE (124)
 57 NURSERY (126)
 60 JOBTABLE (127)
 61 OIFILEB (131)
 62 LOADER1 (132)
 63 CICOMSYS (133)
 64 DEBUG (135)
 65 IOUTILTY (136)
 66 FILESYS1 (137)
 67 FIRMWARESIM3 (3)
 70 FILESYS8 (12)
 71 FIRMWARESIM1 (130)
 72 TRACE0' (140)
 73 CLIB'07 (34)
 74 CLIB'03 (47)
 75 TRACE1' (141)
 76 DISPATCH
 77 IOMDISC0
100 IOTAPE0
101 IOTERM0
102 IOCDRD0
103 IOLPRT0
104 UCOP
105 PFAIL
106 DEVREC
107 LOAD
110 PROGEN
DATE?
```

Figure 6-5. Load Map (Continued)

Table 6-3. Cold Load Error Messages

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.

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.

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.

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 = NOT 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 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

Table 6-3. Cold Load Error Messages (Continued)

*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.

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.

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.

DISC{READ / WRITE / SEEK} ERR 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. 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.

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.

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 6-3. Cold Load Error Messages (Continued)

MOUNT CORRECT VOLUMES OR RELOAD

On a COOLSTART, COLDSTART or UPDATE, not all of the previously-defined volumes were found. You should either mount the correct volumes and start the COOLSTART, COLDSTART or UPDATE over, or you should RELOAD.

OUT OF BOOTSTRAP DISC SPACE

The 30 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 you HP Customer Engineer for assistance.

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 Dialogue).

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), increasing the number of entries in this table (step 4.3 of the Configurator/User Dialogue).

OUT OF MEMORY

The amount of space needed to build the main-memory resident portion of the system, and contain one segment of the Initiator program and its associated table, has exceeded the available core memory. It is possible (although highly unlikely) that a particular memory module has failed and caused the error. Contact the HP Customer Engineer for assistance.

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 Dialogue).

OUT OF SYSTEM DISC SPACE

The virtual memory, directory, and system file disc space required exceeds that available on the system disc. The System Supervisor should reconfigure the system to correct the problem. If another disc with a larger capacity for data is available, use it as the system disc if you do not want to change the tables and other factors which cause the problem.

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 Dialogue).

PREVIOUS RELOAD ABORTED; MUST RELOAD

The last coldload was RELOAD that was aborted. Therefore, this coldload must be a RELOAD.

PREVIOUS TAPE COLD LOAD ABORTED; MUST COLD LOAD FROM TAPE

The last coldload was a COLDSTART or UPDATE that was aborted. Therefore, this coldload must be a COLDSTART, UPDATE or RELOAD.

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.

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.

Table 6-3. Cold Load Error Messages (Continued)

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 is 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.
TAPE PARITY ERROR A parity error was detected while reading the magnetic tape. Try a back-up tape after cleaning the tape heads.
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.
TAPE UNIT WHEN 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.
TIMING ERROR This error, reported by the magnetic tape controller, implies a defective tape controller. Contact your Hewlett-Packard Customer Engineer.
VOLUME TABLE DESTROYED; MUST RELOAD The volume table maintained on the system disc has been overwritten; the system must be reloaded.

BACKING UP DISC FILES ON TAPE

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.

Tapes produced by :SYSDUMP and :STORE are compatible. Those written by :SYSDUMP are used by the MPE Initiator when reloading the system (RELOAD option). Tapes produced by either method are suitable as input to the :RESTORE command (described in Section II) which allows a System Manager or System Supervisor user to restore to disc from a back-up tape 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

The general magnetic tape format of a :STORE command is listed in figure 6-6. Both :STORE and :RESTORE support multi-file and multi-reel files.

The *tape directory* records are 12-word records, 85 records per block. The blocksize is 1024 words. The last record of the tape directory may be shorter than 1024 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.

Each file record on tape is 1024 words long. The last record may be shorter, but will be a multiple of 128 words. The first record of each file contains the *file label* known to the file system.

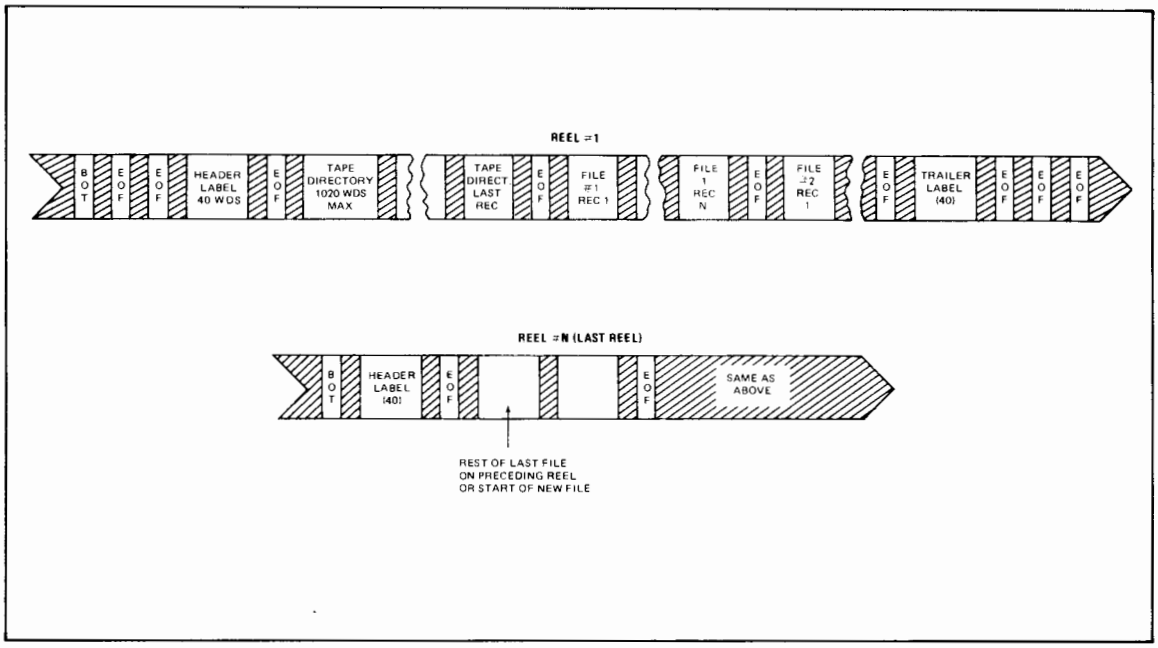


Figure 6-6. :STORE Tape Format

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 the fileset off-line to a magnetic tape unit 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. If the job performing the :STORE/:RESTORE function is aborted by you or at the system console, 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 6-4 lists the messages and their meanings.

The following catastrophic errors abort the :STORE 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 temporary disc files (GOOD, ERROR) used by the :STORE command executor.

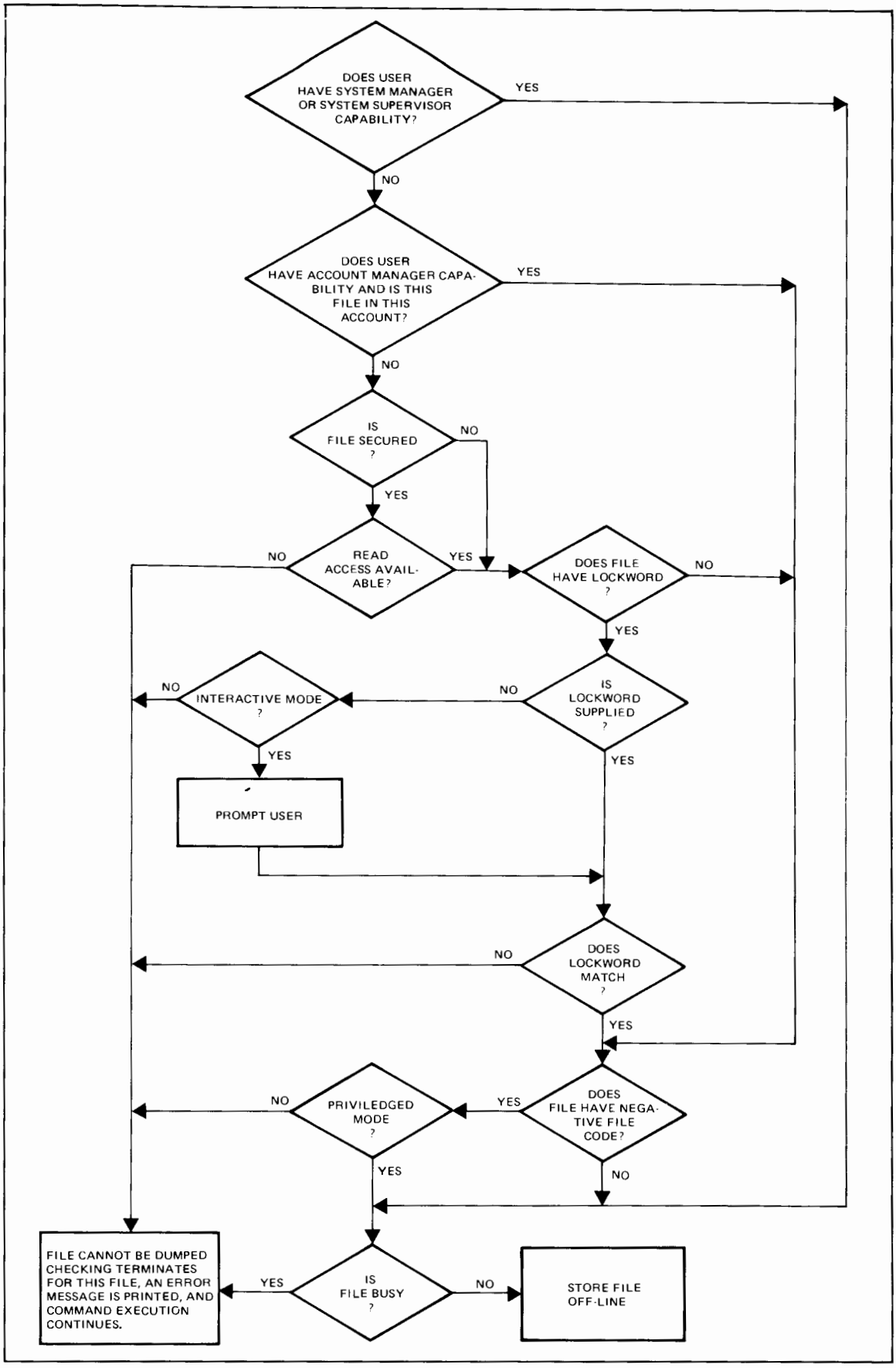


Figure 6-7. Checks for File Dump Eligibility

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 6-4. :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.
BUSY	File is open for output, or is currently being stored or restored.
FILE CODE <0 AND NO PRIVMODE	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 specified incorrectly.
READ ACCESS FAILURE	The user does not have read access to the specified file.
TAPE READ ERROR	A tape read error has occurred on a block other than that containing the file label.
FILE LABEL ERROR	Due to a problem beyond the user's control, the file label is not valid.

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 and group 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 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.

Table 6-5. :RESTORE Command Error Messages

MESSAGE	MEANING
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.

Table 6-5. :RESTORE Command Error Messages (Continued)

MESSAGE	MEANING
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.

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.

Tapes created by the :STORE command and :SYSDUMP command are compatible. Thus, a tape dumped through :SYSDUMP can be used as input for the :RESTORE command. However, :STORE/:RESTORE tapes cannot be used as the first reel of system initiation tapes since the operating system has not been copied to the tape.

CONTROLLING PROCESSES

All processes competing for the central processor access it in an orderly manner, under the direction of MPE. The system places runnable processes in the master scheduling queue in order of their priority. When a process in execution is completed, terminated, or suspended, the MPE Dispatcher searches the master queue for the next process of highest priority, and transfers control to that process.

In the master queue (figure 6-10), a small priority-number (as low as 1) corresponds to a *high priority,* and successively greater priority numbers (up to 255) correspond to successively *lower priorities.* Process priority numbers can only be assigned by users with the System Supervisor or Privileged Mode optional capability. Other (standard) users reference the master queue as a set of logical areas called *subqueues.* Each subqueue corresponds to a *priority class* (AS, BS, CS, DS, or ES), a *scheduling type* (linear or circular), and a *range of priority numbers* normally assigned to processes in that subqueue.

In linear subqueues (AS and BS), processes are assigned fixed priorities and access the central processor as they need it, with the process of highest priority obtaining access first. In circular subqueues (CS, DS, and ES), the priorities of processes are dynamically adjusted by an amount depending on their priority classes; the number of milliseconds allowed a running process before this adjustment takes place is called the *time quantum.* Expiration of the quantum may cause the process to lose control of the central processor if the adjustment results in assignment of a higher priority to another process. In circular subqueues, the priority of a particular process tends to fall as the central processor is used but rises while the process is waiting for the central processor; this helps ensure optimum use of equipment and fair-sharing of the system. The standard subqueues (priority classes) are as follows:

AS

Is a linear subqueue containing processes of very high priority. Its priority range is 30-99 and it is presently used only by MPE.

Scheduling Type: Linear
Priority Range: 30-99

BS

Is a linear subqueue containing processes of very high priority. It is accessible to users having MAXPRI = BS. Normally priority for a BS process is 100. However, by specifying a rank greater than 0 in the CREATE or GET PRIORITY intrinsics, the process may be set in the master queue at the lesser of the following two values: (100 + *rank*) or 149.

Scheduling Type: Linear
Priority Range: 100-149

CS

Is a circular subqueue generally devoted to interactive sessions. A CS process which uses its quantum of CPU will be lowered in priority, but not below the C Subqueue Priority Limit, called cpri (which may be set in the :SYSDUMP dialogue or the :QUANTUM command).

Scheduling Type: Circular
Priority Limits: tpri-cpri

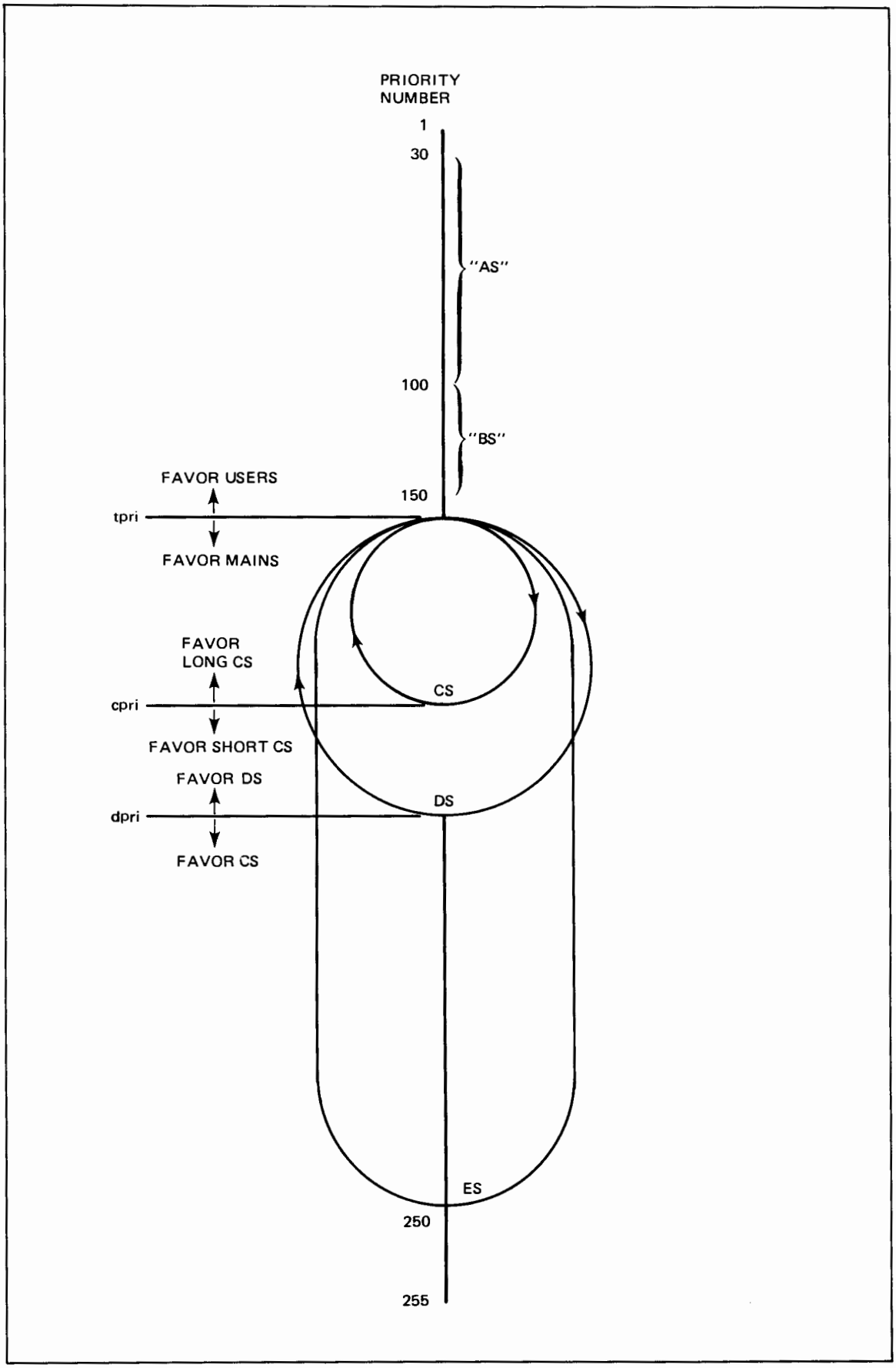


Figure 6-10. MPE Master Queue Structure

DS

Is a circular subqueue generally devoted to batch jobs. A DS process which uses its quantum of CPU will be lowered in priority more rapidly than a CS process, but not below the D Subqueue Priority Limit, or dpri (which may be set in the :SYSDUMP dialogue or the :QUANTUM command).

Scheduling Type: Circular
Priority Limits: *tpri-dpri*

ES

Is a circular subqueue generally used for so-called "idle" processes. When an ES process consumes a quantum of CPU, its priority is set to 250. Such a process will have a minimal impact on the performance of processes in other subqueues.

Scheduling Type: Circular
Priority Limits: *tpri*-250

The priority class of a process can be specified by the normal user with standard or optional MPE capabilities. In the two-character string that comprises a priority-class reference, the first character refers to the location of a subqueue within the master queue (in alphabetical order) and the second character specifies whether the logical area is the subqueue itself (S) or the portion of the master queue (M) that immediately follows the subqueue.

Priority numbers range from 1 to 255 inclusively, with 1 denoting the highest priority. Only the System Supervisor user can specify the relative ranks of processes within a linear subqueue or specify the priority ranges of circular subqueues. Only the Privileged Mode user can request a portion of the master queue or specify an absolute priority number for a process.

A System Supervisor user can control and monitor the master queue in the following ways:

- Modify the quantum for the circular subqueues (:QUANTUM).
- Change the priority limits for the circular subqueues (:QUANTUM).
- Display the master queue and subqueues (:SHOWQ).

These functions may be performed on-line in either sessions or batch jobs during normal system operation.

In addition to monitoring the master queue, a System Supervisor user can also control the scheduling of individual processes within linear subqueues. He does this with the *rank* parameter of the CREATE and GET PRIORITY intrinsics. (See the *MPE Intrinsics Reference Manual.*) The *rank* parameter is available only to System Supervisors, and is defined as follows:

rank

The relative rank of the process within a linear subqueue. Processes in a linear subqueue are linked together, defining an implicit order (1, 2, 3, and so forth). The *rank* parameter defines the *relative* location of a process in the chain; it is not kept as a fixed attribute of the process. If omitted, the process is scheduled as the last of the chain.

PERMANENTLY ALLOCATING PROGRAMS AND PROCEDURES

A user with System Supervisor capability can enhance system efficiency by storing program files (sets of one or more segments) or procedures (individual segments) permanently in virtual memory. This is, in effect, a permanent allocation of the programs or procedures so that they always reside in virtual memory 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 inserting and withdrawing them from virtual memory. 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 remove from virtual memory 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 and assigned virtual memory space (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 system-wide 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 (≤ 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.

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.

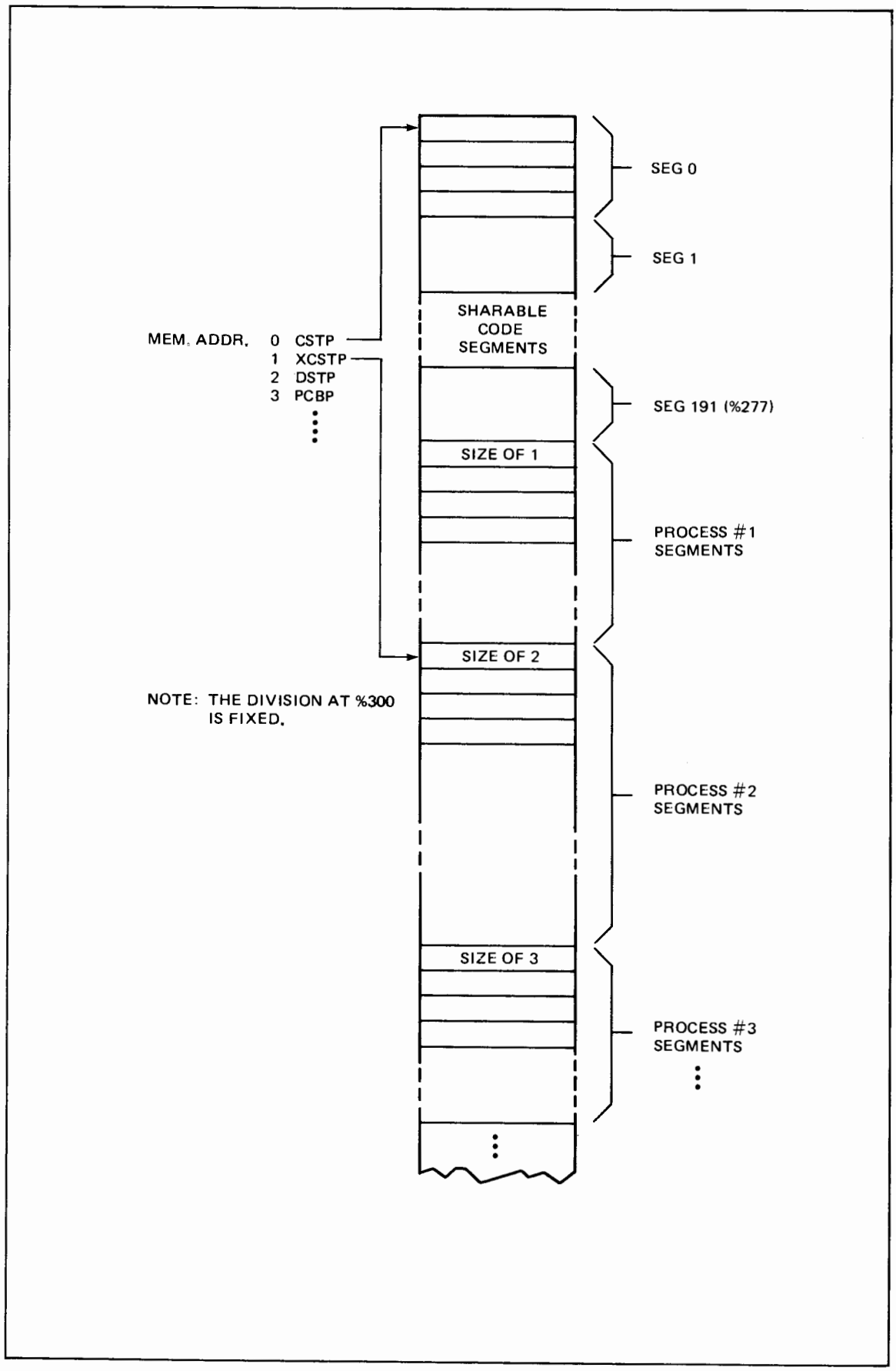
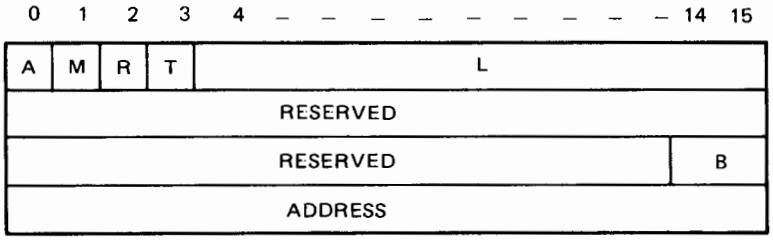


Figure 6-11. Code Segment Table Format

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 process (in figure 6-11, process 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 = for an n word code segment, L = $n/4$ and PL = PB + (4 * L) – 1.

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.

B = Bank address, and points to the bank (if core resident) in which the segment resides.

ADDRESS = Absolute address of PB within B 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.

USING THE LOGGING SYSTEM

In addition to the Accounting System, MPE provides another facility that can be used for monitoring the use of certain resources by accounts, groups, and users — the Logging System. While the Accounting System is managed by System and Account Manager users, the Logging System is managed by System Supervisor users. Like the Accounting System, the Logging System can be used for billing or for simply obtaining an overview of system usage. Unlike the Accounting System, however, the Logging System 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 closing.
- System shut-down.
- Spooling.
- Line close/line disconnection.
- System I/O error monitoring.

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 user 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.

LOG FILES

The Logging System is activated during system configuration, through the Configurator/user dialogue (Steps 6 through 6.4) discussed earlier. These steps allow you to:

- Enable or disable logging. (This option is also available during coldloading in Step 5.5 of the user/Initiator 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 logging is enabled for the first time, the first log file is created and opened on disc, and records are written to it by the Logging System 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 user, 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.

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 6.3 and 6.4, respectively, of the Configurator/user dialogue.

The *block size* is designated 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 always is 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:

$$\text{Sectors} = \left[\left(\frac{\text{Log File Size}}{16}\right) \times \left(\text{Log Record Size}\right)\right]$$

FILE SECURITY. — Because log files are files belonging to and created by the MPE Logging System, their creator, by implication, is the original System manager user (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, the System Manager user can re-name the file (perhaps adding a *lockword*), and can re-define its security provisions. *Thus, the System Manager user controls access to the current log file, but the System Supervisor user 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 user 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 user 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 the Logging System 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 user 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 user 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.

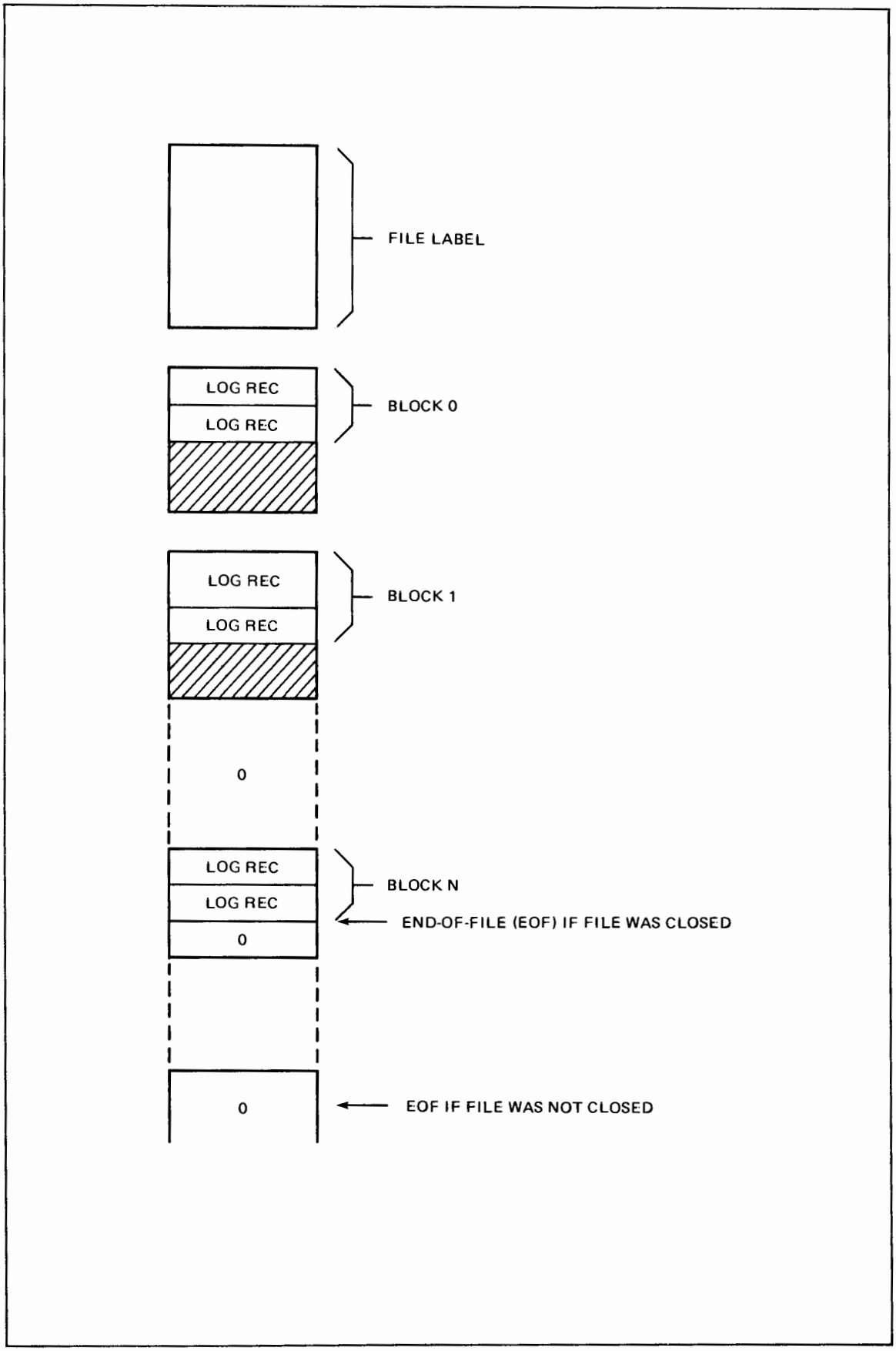


Figure 6-12. Log File Format

LOG RECORDS. — Log records are written to the log file by MPE. This is done by calls to a special log intrinsic from the 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.

Twelve types of log records can be entered; see table 6-6.

Table 6-6. Log Record Types

TYPE NO.	TYPE
0	Log Failure Record
1	Head Record
2	Job Initiation Record
3	Job Termination Record
4	Process Termination Record
5	File Close Record
6	Shut-Down Record
7	Power Failure Record
8	Spooling Log Record
9	Line Disconnection
10	Line Close
11	I/O Error

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.

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 12 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

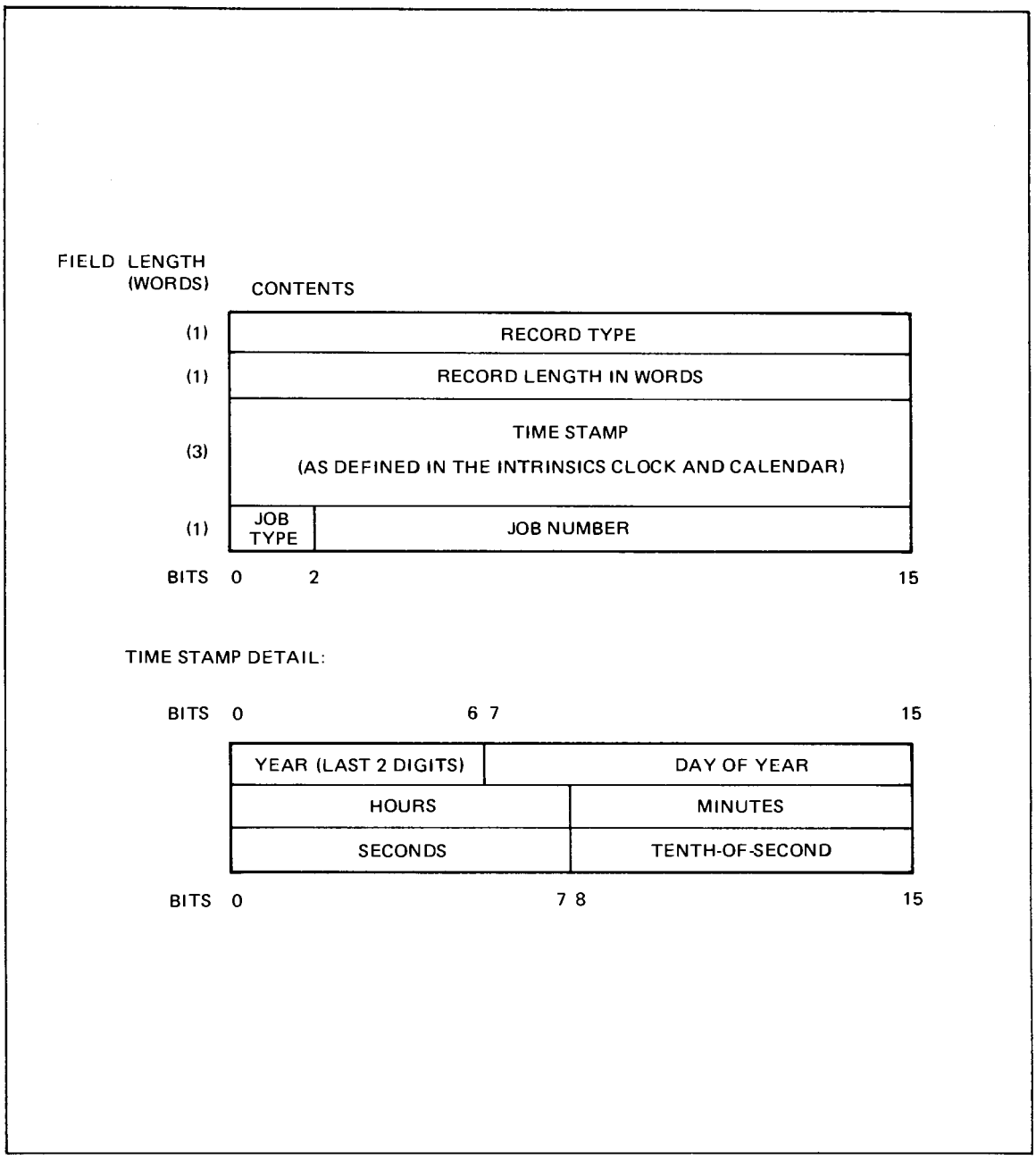
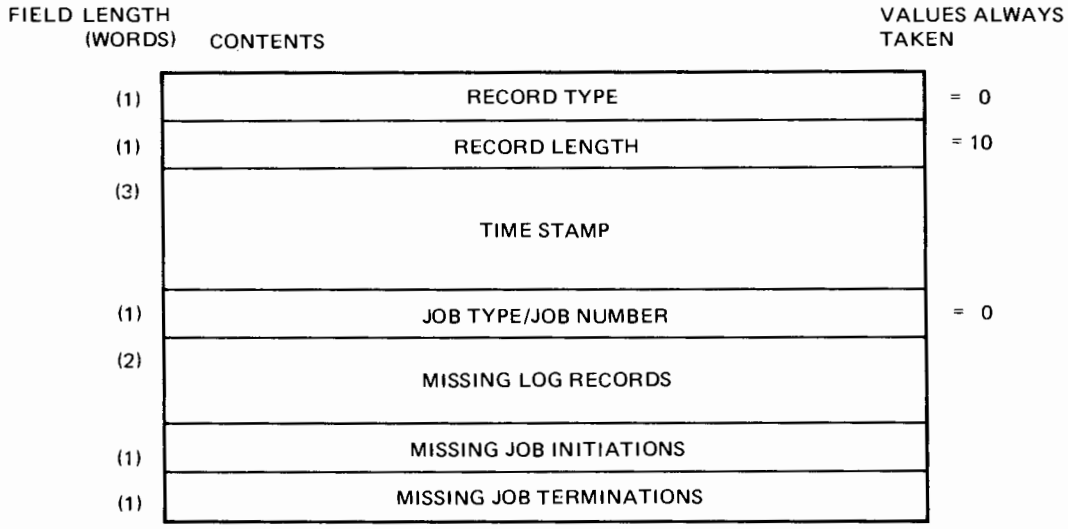


Figure 6-13. Log Record Heading Format

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.

Beyond the heading, the information in each individual record depends on the type of record and the particular event logged.

The general formats of all 12 types of records are described in figures 6-14 through 6-25.



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	
SYSTEM NUMBER	(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.

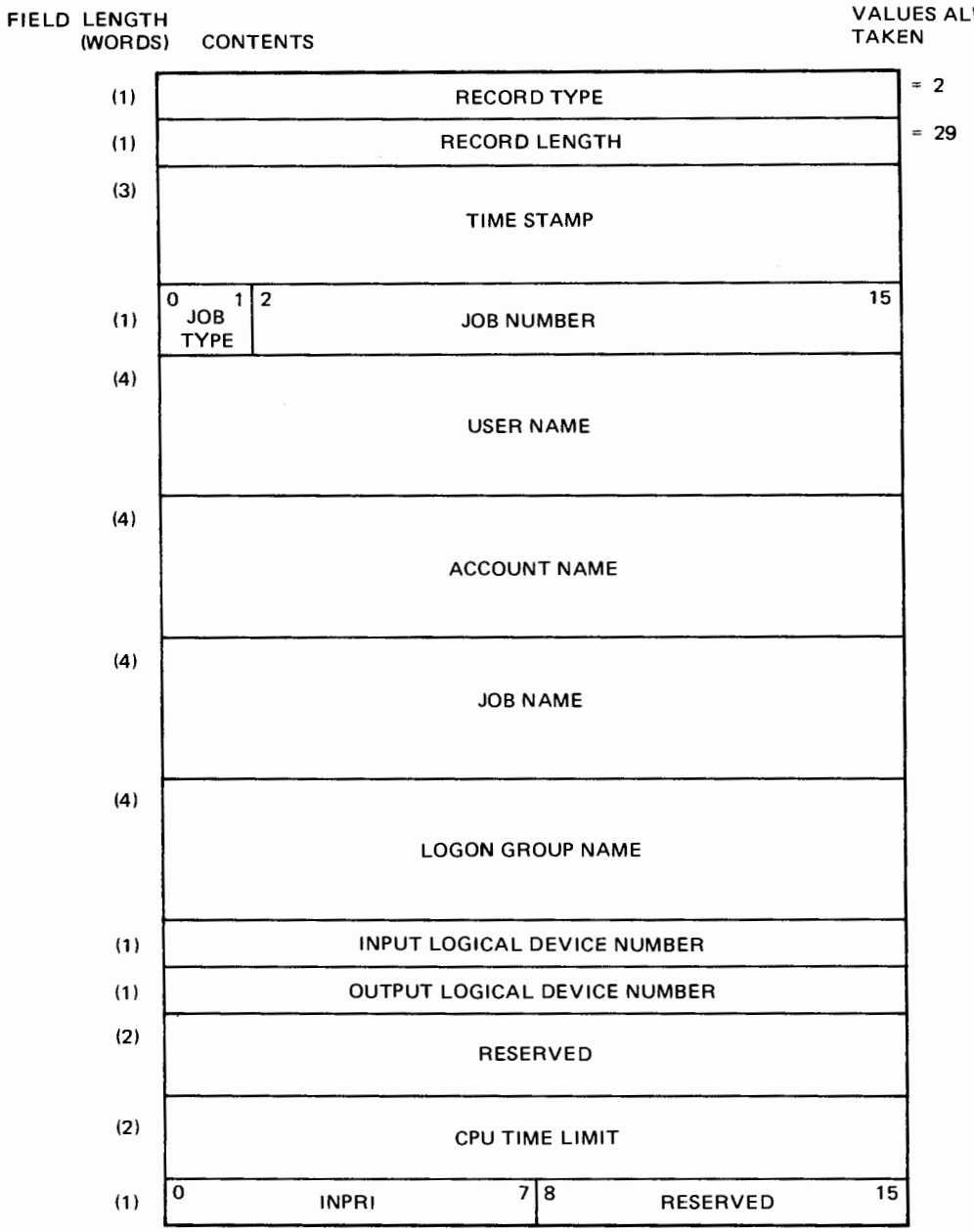
TRL SIZE (1 word) is the number of entries in the Time Request List.

ICS SIZE (1 word) is the number of words in the Interrupt Control Stack.

MAX # OF JOBS (1 word) is the maximum number of jobs/sessions allowed in the system.

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 log-on 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.
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, left-justified)	is the job selection priority as defined on the :JOB card.

Figure 6-16. Job Initiation Record (Type 2) (Continued)

FIELD LENGTH (WORDS)	CONTENTS	VALUES ALWAYS TAKEN
(1)	RECORD TYPE	= 3
(1)	RECORD LENGTH	= 12
(3)	TIME STAMP	
(1)	0 1 JOB TYPE / 2 15 JOB NUMBER	
(1)	MAXIMUM PRIORITY	
(1)	NUMBER OF CREATIONS	
(2)	CPU TIME IN SECONDS	
(2)	ELAPSED TIME IN MINUTES	

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)

FIELD LENGTH (WORDS)	CONTENTS	VALUES ALWAYS TAKEN
(1)	RECORD TYPE	= 4
(1)	RECORD LENGTH	= 11
(3)	TIME STAMP	
(1)	0 1 JOB TYPE / 2 15 JOB NUMBER	
(1)	#OF PROGRAM FILE SEGMENTS	
(1)	#OF SL SEGMENTS (NON MPE)	
(1)	MAXIMUM STACK SIZE EVER	
(1)	MAXIMUM DATA SEGMENT SIZE EVER	
(1)	CUMULATIVE TOTAL OF VIRTUAL STORAGE	

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.

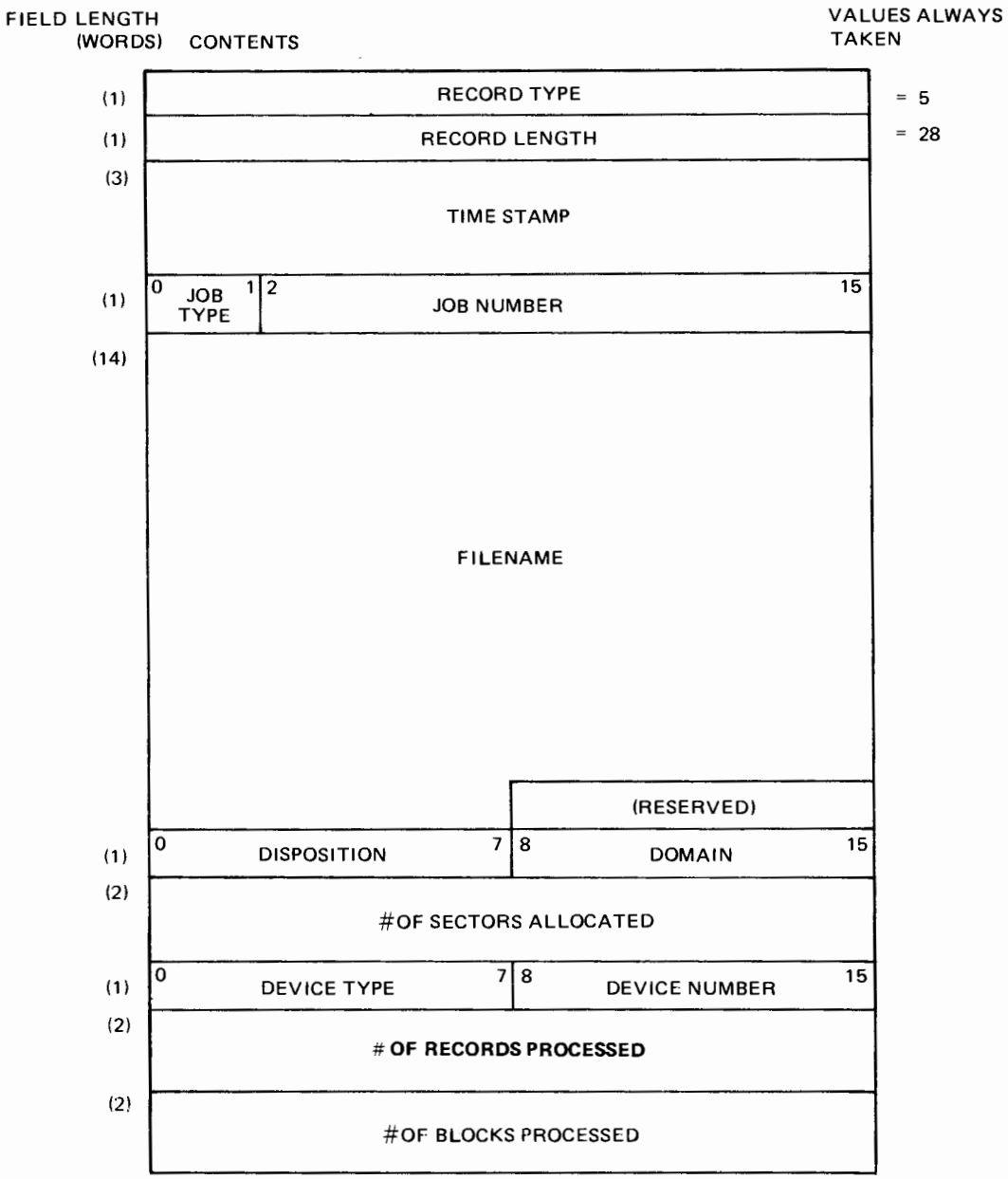
NUMBER OF NON-MPE SL SEGMENTS (1 word) is the number of segments from the segmented library (excluding MPE), loaded on behalf of the process.

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, with the last byte containing a blank.

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.)

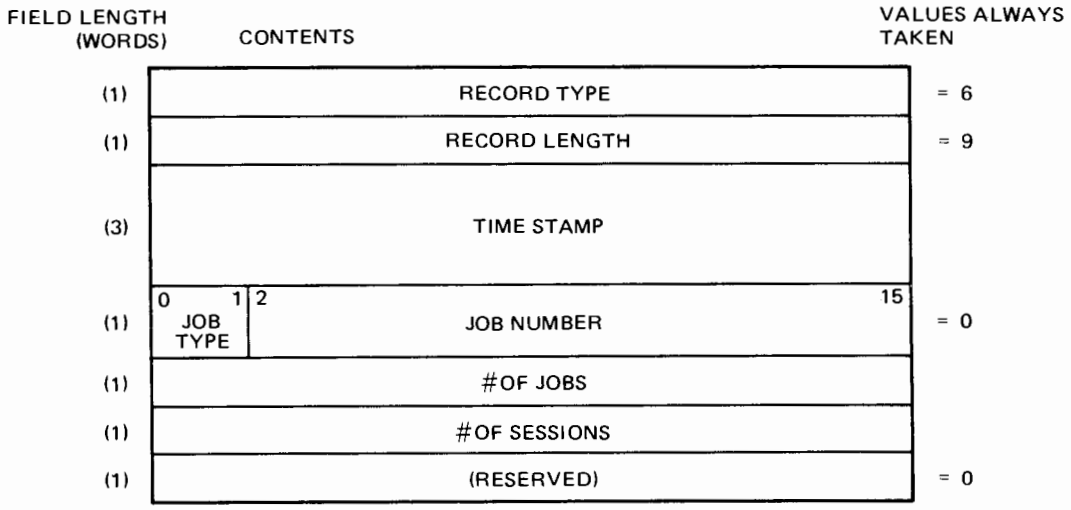
DEVICE TYPE (1 byte) is the device type of the device on which the file resides, as returned by the FGETINFO intrinsic.

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)



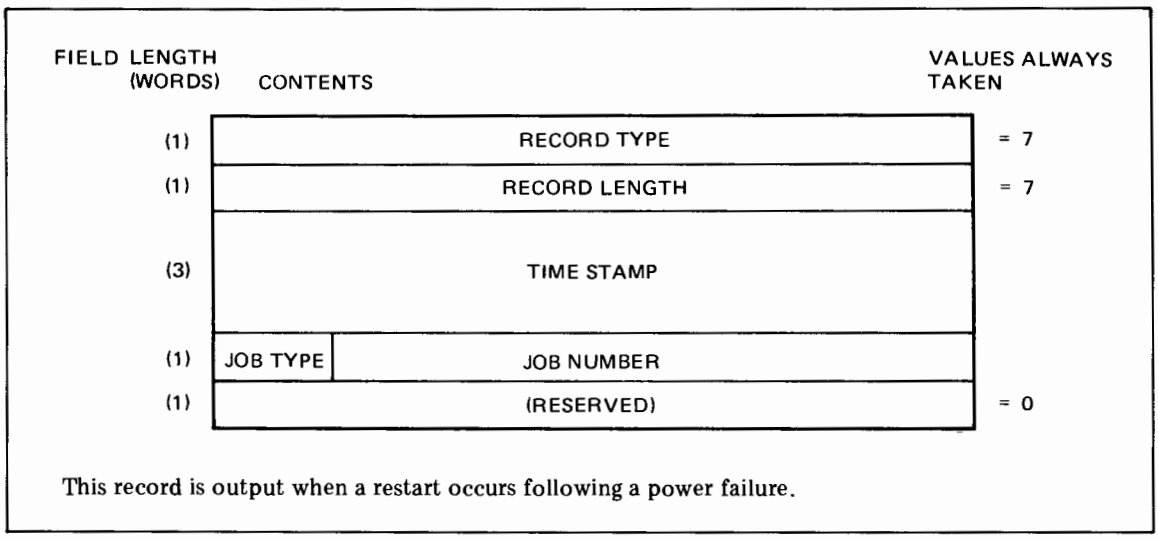
This record is output when the system is shut down (by the =SHUTDOWN console command. See the MPE Console Operator's Guide.).

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.

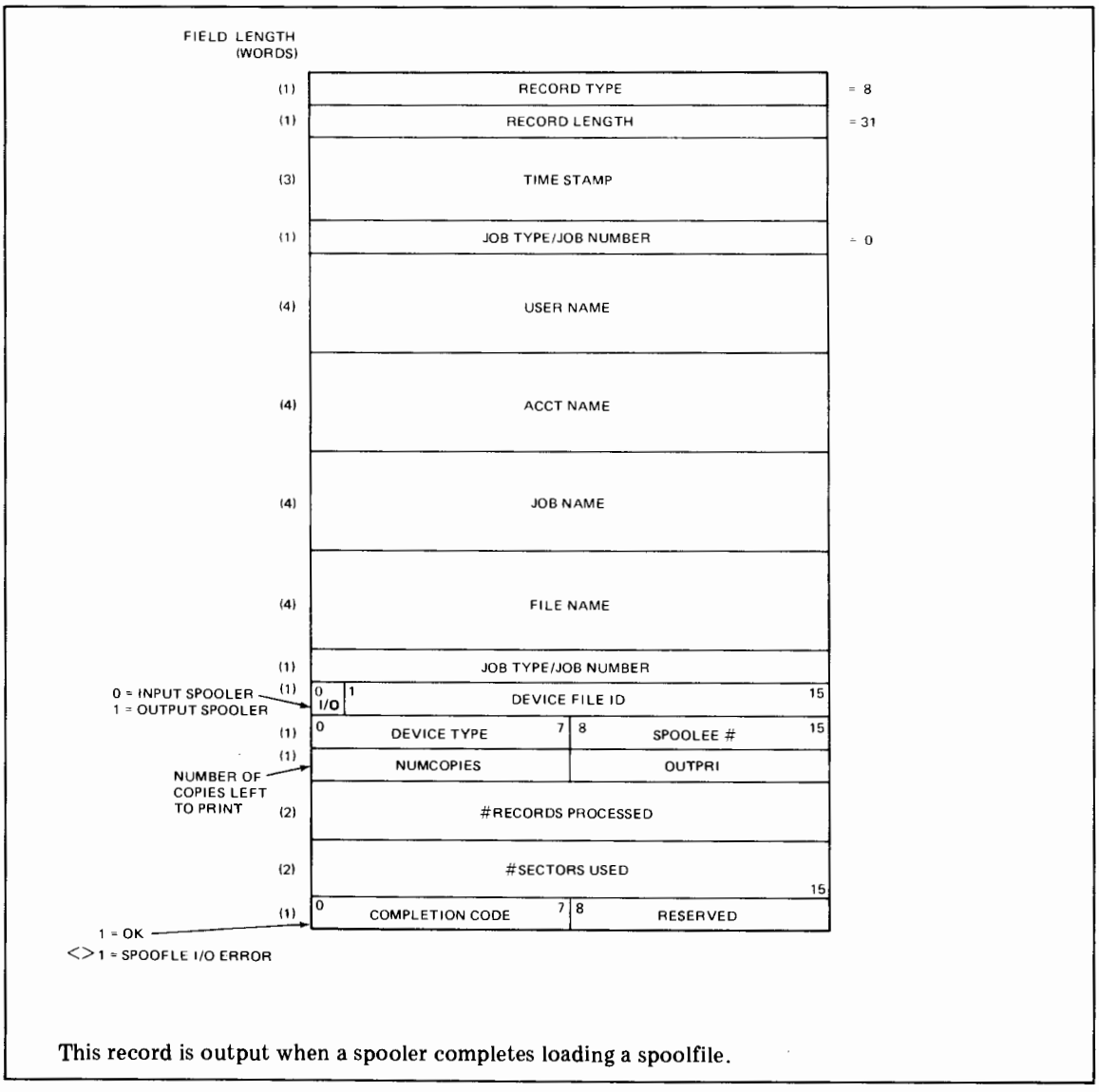
NUMBER OF SESSIONS ON SYSTEM (1 word) is the number of sessions on the system when the command took effect.

Figure 6-20. Shut-Down Record (Type 6)



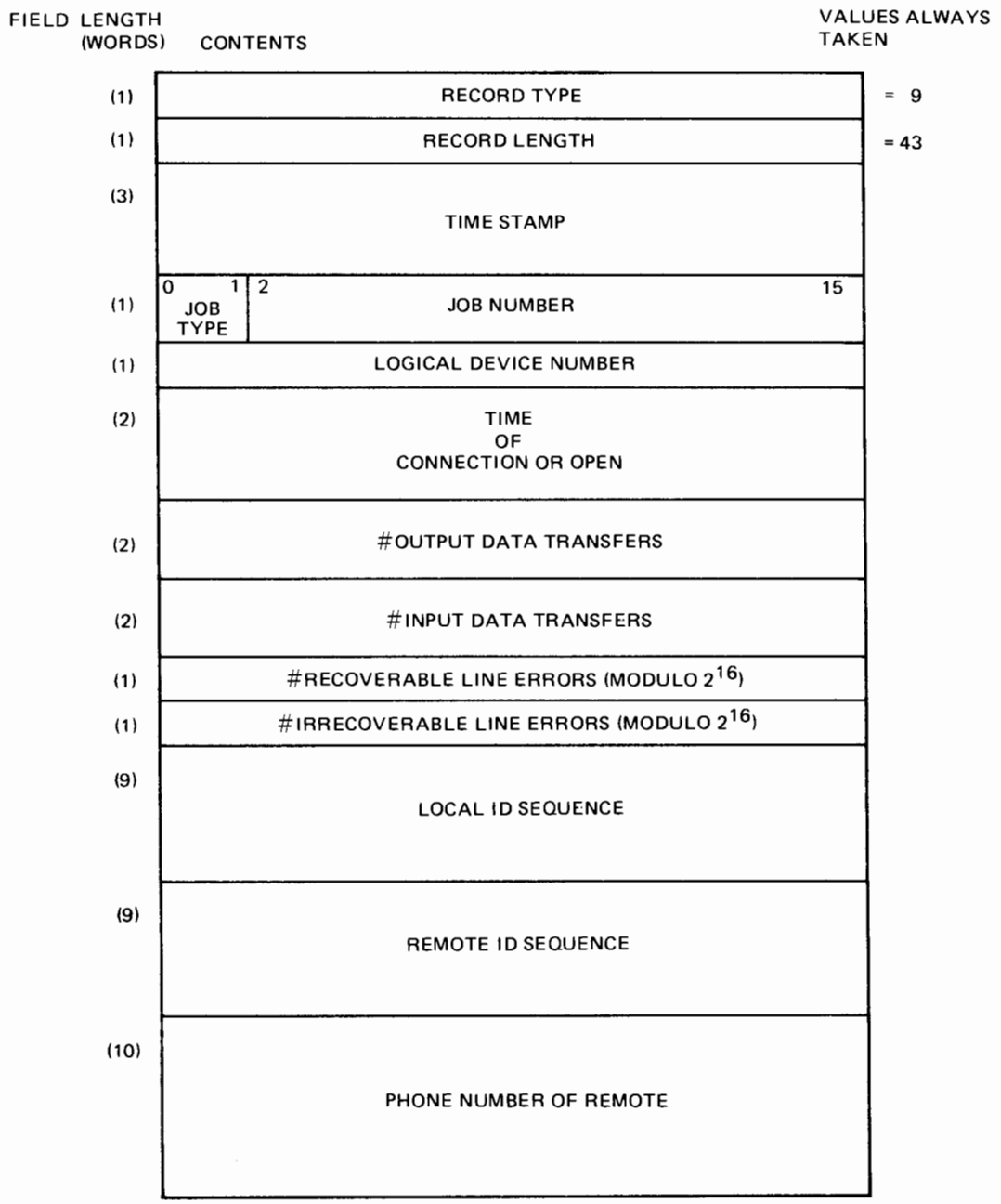
This record is output when a restart occurs following a power failure.

Figure 6-21. Power Fail Record (Type 7)



This record is output when a spooler completes loading a spoolfile.

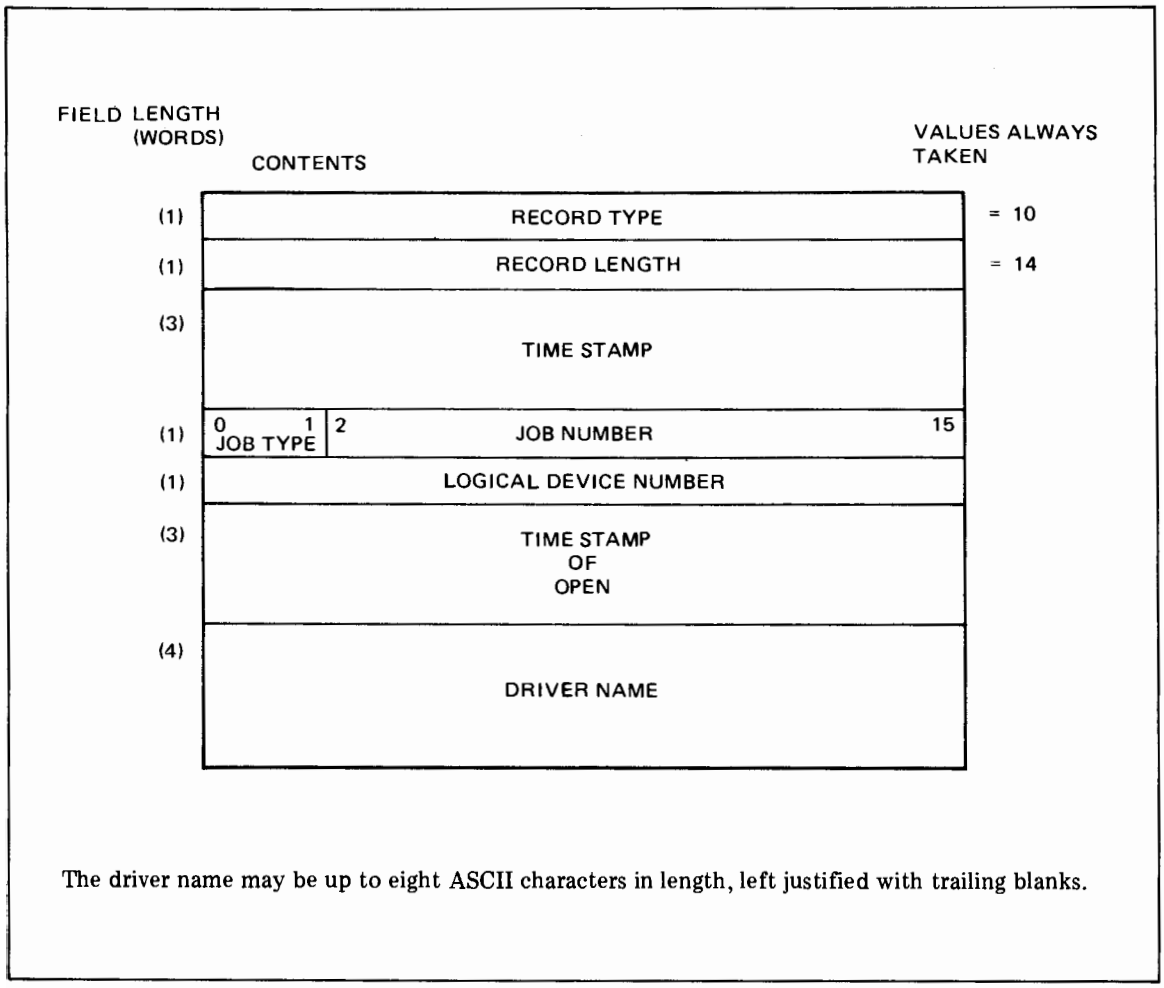
Figure 6-22. Spooler's Spoolfile Done Record (Type 8)



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).

The phone number is significant only if the local system (MPE) performed the dialing for an outgoing call. The first byte is the length of the phone number. The remainder is the actual phone number represented in ASCII. (Up to 20 characters, left justified.)

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)

FIELD LENGTH WORDS	CONTENTS	VALUES ALWAYS TAKEN
(1)	RECORD TYPE	= 11
(1)	RECORD LENGTH	= 20
(3)	TIME STAMP	
(1)	0 JOB TYPE 1 | 2 JOB NUMBER 15	
(1)	HARDWARE ERROR STATUS	
(1)	A S D Y W B C F M P L R T G	
(1)	LOGICAL DEVICE NUMBER	
(1)	MISCELLANEOUS (DRIVER DEFINED) DATA	
(1)	TARGET DATA SEGMENT (SEE DISCUSSION BELOW)	
(1)	TARGET DATA ADDRESS WITHIN DATA SEGMENT	
(1)	DRIVER FUNCTION CODE	
(1)	TRANSMISSION COUNT (NEGATIVE IF IN BYTES)	
(1)	DRIVER DEFINED PARAMETER #1	
(1)	DRIVER DEFINED PARAMETER #2	
(1)	0 PROCESS PCB NUMBER 7 | STAT QUALIFIER | STAT	
(1)	4 SUBTYPE 7 | 8 DEVICE TYPE 15	
(1)	0 UNIT NUMBER 7 | 8 DRT 15	

The following is an explanation of the bits in the word following the hardware error status 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.

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)

CONSOLE MESSAGES FOR LOG FILES. — Log file errors and status are reported to the system console through messages of the following format:

ST/hh:mm/message

where

hh	The hour of the day.
mm	The minute of the hour.
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 6-7. Logging is disabled until the next cold load or reload.

LOG FILE NUMBER x ERROR #nn. LOGGING SUSPENDED

A recoverable error occurred, where *nn* is the error number as listed in table 6-7. Logging is temporarily suspended pending a :RESUMELOG command.

LOG FILE NUMBER x LOGGING RESUMED

A :RESUMELOG Command was successfully executed.

Table 6-7. 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

DRIVER NAMES, TYPES, SUB-TYPES AND SIZES

APPENDIX A

DEVICE	PART NO.	DRIVER NAME	TYPE	SUB-TYPE	RECORD WIDTH (Decimal Words)
Asynchronous Terminal Controller	30032B	IOTERM0	16		36
Hardwired Terminal				0*	
Terminal Interfaced Over 103A Modem				1	
Terminal Interfaced Over Asynchronous Half-Duplex Modem with Data-Rate Select ON.				2	
Terminal Interfaced Over Asynchronous Half-Duplex Modem with Data-Rate Select OFF.				3	
Nine-channel Magnetic Tape Unit	**7970B**	IOTAPE0	24	0	128
Fixed-Head Disc	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	
Cartridge Disc (7905)	**13180B**	IOMDISC1	0		128
Removable Platter				4	
Fixed Platter				5	
Cylinder Mode				6	
System Disc Mode				7	
Disc File	30102A	IOMDISC0	0	3	128
Card Reader	30106A	IOCDRD0	8	0	40
Paper Tape Reader	30104A	IOPTRD0	9	0	40
Paper Tape Punch	30105A	IOPTPN0	34	0	128
Line Printer	30108A 30109A **2607** **2613** **2618**	IOLPRT0	32	0 1 2	66

*These terminals should be configured with Sub-Type = 1 when hardwired: ASR 37, Memorex 1240.

DEVICE	PART NO.	DRIVER NAME	TYPE	SUB-TYPE	RECORD WIDTH (Decimal Words)
Plotter	30226A	IOPLOT0	35		128
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	
Printing Reader/Punch	30119A	IOPRPN0	20	0 1	40 48
Programmable Controller (UI)	30361A	IOREM0	23	0	128
Synchronous Single-Line Controller	30055A	CSSBSC0	18		128
ASCII/Switched				0	
ASCII/Non-switched				1	
EBCDIC/Switched				2	
EBCDIC/Non-switched				3	
Hardwired Serial Interface	30360	CSHBSC0	19	0	128
Distributed Systems Driver		IODS0	41	0	128
Distributed Systems Dummey Terminal		IODSTRM0	16	0	36

DRIVER CONFIGURATION DATA

APPENDIX B

For the following Hewlett-Packard-supported devices and drivers, the user should input the information described below in Steps 3.3.4 through 3.3.12 and Step 10.1 of the Configurator/ user dialogue.

ASYNCHRONOUS TERMINAL CONTROLLER DRIVER (IOTERM0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	16
3.3.5	SUB TYPE?	(0, 1, 2, or 3 — Appendix A)
3.3.6	TERM TYPE?	(Varies with terminal)
3.3.7	REC WIDTH?	(Varies with terminal)
3.3.8	OUTPUT DEVICE?	(No. of corresponding listing device)
3.3.9	ACCEPT JOBS/SESSIONS?	YES
3.3.10	ACCEPT DATA?	YES
3.3.11	INTERACTIVE?	YES
3.3.12	DUPLICATIVE?	YES
3.3.13	INITIALLY SPOOLED?	NO
3.3.15	DRIVER NAME?	IOTERM0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOTERM0, <prepared file name>

NINE-CHANNEL MAGNETIC TAPE UNIT DRIVER (IOTAPE0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	24
3.3.5	SUB TYPE?	0
3.3.7	REC WIDTH?	128
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	YES or NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	YES or NO
3.3.14	INPUT OR OUTPUT?	IN (asked only if initially spooled)
3.3.15	DRIVER NAME?	IOTAPE0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOTAPE0, <prepared file name>

FIXED-HEAD DISC DRIVER (IOFDISC0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	1
3.3.5	SUB TYPE?	(1, or 2)
3.3.7	REC WIDTH?	128
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	NO
3.3.15	DRIVER NAME?	IOFDISC0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOFDISC0, <prepared file name>

7900A CARTRIDGE DISC DRIVER (IOMDISC0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	0
3.3.5	SUB TYPE?	(0, 1, or 2)
3.3.7	REC WIDTH?	128
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	NO
3.3.15	DRIVER NAME?	IOMDISC0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOMDISC0, <prepared file name>

7905A CARTRIDGE DISC DRIVER (IOMDISC1)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	0
3.3.5	SUB TYPE?	(4, 5, 6, or 7)
3.3.7	REC WIDTH?	128
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	NO
3.3.15	DRIVER NAME?	IOMDISC1
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOMDISC1 <prepared file name>

DISC FILE DRIVER (IOMDISC0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	0
3.3.5	SUB TYPE?	3
3.3.7	REC WIDTH?	128
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	NO
3.3.15	DRIVER NAME?	IOMDISC0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOMDISC0, <prepared file name>

CARD READER DRIVER (IOCDRD0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	8
3.3.5	SUB TYPE?	0
3.3.7	REC WIDTH?	40
3.3.8	OUTPUT DEVICE?	(Device desired)
3.3.9	ACCEPT JOBS/SESSIONS?	YES
3.3.10	ACCEPT DATA?	YES
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	YES or NO
3.3.15	DRIVER NAME?	IOCDRD0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOCDRD0, <prepared file name>

LINE PRINTER DRIVER (IOLPRT0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	32
3.3.5	SUB TYPE?	(0, 1, or 2)
3.3.7	REC WIDTH?	66
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	YES or NO
3.3.15	DRIVER NAME?	IOLPRT0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOLPRT0, <prepared file name>

PAPER TAPE READER DRIVER (IOPTRD0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	9
3.3.5	SUB TYPE?	0
3.3.7	REC WIDTH?	40
3.3.8	OUTPUT DEVICE?	(Device desired)
3.3.9	ACCEPT JOBS/SESSIONS?	YES
3.3.10	ACCEPT DATA?	YES
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	NO
3.3.15	DRIVER NAME?	IOPTRD0
10.1	ENTER PROGRAN NAME, REPLACEMENT FILE NAME?	IOPTRD0, <prepared file name>

PAPER TAPE PUNCH DRIVER (IOPTPN0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	34
3.3.5	SUB TYPE?	0
3.3.7	REC WIDTH?	128
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	NO
3.3.15	DRIVER NAME?	IOPTPN0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOPTPN0, <prepared program file name>

PLOTTER DRIVER (IOPLOT0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	35
3.3.5	SUB TYPE?	(0 through 7 — see Appendix A)
3.3.7	REC WIDTH?	128
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	YES or NO
3.3.15	DRIVER NAME?	IOPLOT0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOPLOT0, <prepared program file name>

PRINTING READER/PUNCH (IOPRPN0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	20
3.3.5	SUB TYPE?	0
3.3.7	REC WIDTH?	40
3.3.8	OUTPUT DEVICE?	(desired device)
3.3.9	ACCEPT JOBS/SESSIONS?	YES
3.3.10	ACCEPT DATA?	YES
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	YES or NO
3.3.14	INPUT OR OUTPUT?	IN or OUT
3.3.15	DRIVER NAME?	IOPRPN0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOPRPN0, <prepared program file name>

PROGRAMMABLE CONTROLLER (UI) (IOREM0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	23
3.3.5	SUB TYPE?	0
3.3.7	REC WIDTH?	128
3.3.8	OUTPUT DEVICE?	0
3.3.9	ACCEPT JOBS/SESSIONS?	NO
3.3.10	ACCEPT DATA?	NO
3.3.11	INTERACTIVE?	NO
3.3.12	DUPLICATIVE?	NO
3.3.13	INITIALLY SPOOLED?	NO
3.3.15	DRIVER NAME?	IOREM0
10.1	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	IOREM0, <prepared program file name>

SYNCHRONOUS SINGLE-LINE CONTROLLER (CSSBSC0)

CONFIGURATOR STEP NO.	CONFIGURATOR OUTPUT	USER RESPONSE
3.3.4	TYPE?	18
	SUB TYPE?	(0 through 3 — see Appendix A)
3.3.6.1	PROTOCOL?	1
3.3.6.2	LOCAL MODE?	(Mode desired.)
	•	
	•	
	•	
3.3.6.16	DRIVER CHANGEABLE?	NO
3.3.6.17	DRIVER OPTIONS?	0
3.3.6.18	DRIVER NAME?	CSSBSC0
3.3.6.19	CONTROL LENGTH?	0
	•	
	•	
	•	
10	ENTER PROGRAM NAME, REPLACEMENT FILE NAME?	CSSBSC0, < prepared program file name>

RECOMMENDED PARAMETER SETTINGS

APPENDIX C

Notes for table, next two pages:

1. N = Number of configured devices.

2. The values in the I/O queue are based on installations supporting one spooled line printer and one spooled card reader. If your installation is different, add or subtract 15 for each spooled output device and 10 for each spooled input device. However, the absolute minimum recommended is 20 regardless of configuration.

3. Values should be based on 6 times the number of concurrent active terminal users anticipated for the system.

4. SPOOLFILE kilosectors is dependent on disc configuration (number of discs and size of discs) and planned spooling load.

5. The maximum size of a spoolfile is 32 times the number of sectors per extent. Choose the number of sectors per extent according to the size of your largest anticipated spoolfile. 384 is adequate for most applications.

NOTE

Each sector contains 128 words (256 bytes).

CONFIGURATOR QUESTION \ MEMORY SIZE IN KILOWORDS	64		96	128	160	192	224	256	UNIT OF MEASURE
MAX # OF OPEN SPOOLFILES = XXX.?	20		20	20	**20**	**20**	**20**	**20**	— —
CST = XXX.?	128		128	128	192	192	192	192	ENTRIES
EXTENDED CST = XXXX.?	**192**		512	768	1024	1024	1280	1280	ENTRIES
DST = XXXX.?	192		256	256	**256**	384	**384**	**384**	ENTRIES
PCB = XXX.?	48		64	96	96	96	128	128	ENTRIES
I/O QUEUE = XXX.?	40		50	64	64	80	96	128	ENTRIES
TERMINAL BUFFERS = XXX.?	**40**		**92**	**92**	**128**	**128**	**192**	192	BUFFERS
SYSTEM BUFFERS = XXX.?	8		**8**	**8**	**8**	**8**	**8**	**8**	BUFFERS
MEMORY MANAGEMENT TABLE = XXXX.?	48		96	128	192	256	384	384	ENTRIES
ICS = XXXX.?	512		512	512	512	512	512	512	WORDS
UCOP REQUEST QUEUE = XXX.?	32		40	40	40	48	48	48	ENTRIES
TIMER REQUEST LIST = XXX.?	16		24	24	32	40	48	56	— —
BREAKPOINT TABLE = XXX.?	12		18	24	32	48	64	64	ENTRIES
# RINS MIN = 5, MAX = XXXX.?	48		48	54	56	60	64	64	— —
# OF GLOBAL RINS USED = 0, MAX = XXXX.?	32		32	40	42	46	50	50	— —
# OF SECONDS TO LOGON = XXX.?	120		120	120	120	120	120	120	SECONDS
MAX # OF CONCURRENT SESSIONS = XXX.?	16		24	30	36	40	50	60	— —
MAX # OF CONCURRENT RUNNING JOBS = XXX.?	1		3	4	5	6	7	8	— —
DEFAULT JOB CPU TIME LIMIT = XXXXX.?	0		0	0	0	0	0	0	**SECONDS**
LOG FILE RECORD SIZE (SECTORS) = X.?	2		2	2	2	2	2	2	SECTORS
LOG FILE SIZE (RECORDS) = XXXXX.?	1023		1023	1023	1023	1023	1023	1023	— —
VIRTUAL MEMORY = XXXXX.?	3072		4096	5120	5120	6144	8192	8192	SECTORS
DIRECTORY USED = XXX, MIN = 384 MAX = XXXX.?	512		768	768	768	1024	1024	1024	SECTORS
MAX #SPOOLFILES KILOSECTORS = XXXXXX.?	**128**		128	128	**128**	**128**	**128**	**128**	KILOSECTORS
# SECTORS PER SPOOLFILE EXTENT = XXXX.?	384		384	384	384	384	384	384	SECTORS
TIME QUANTUM = XXXXX.?	**500**		**500**	**500**	**500**	**500**	**500**	**500**	MILLISECONDS
PRIORITY-TERMINAL I/O BOUND JOBS = XXX.?	152		152	152	152	152	152	152	— —
PRIORITY-NORMAL JOBS = XXX.?	**160**		**160**	**160**	**160**	**160**	**160**	**160**	— —
PRIORITY-CPU BOUND JOBS = XXX.?	**180**		**180**	**180**	**180**	**180**	**180**	**180**	— —
MAX # OF CONCURRENT RUNNING PROGRAMS = XXX.?	32		40	40	48	48	64	64	PROGRAMS
MAX CODE SEG SIZE = XXXXX.?	4096		5120	5120	6144	6144	8192	8192	WORDS
MAX # OF CODE SEGMENTS/PROCESS = XXX.?	40		40	50	50	50	64	64	— —
MAX STACK SIZE = XXXXX.?	31232		31232	31232	31232	31232	31232	31232	WORDS
MAX EXTRA DATA SEG SIZE = XXXXX.?	4896		6144	8192	8192	16384	16384	31232	WORDS
MAX # EXTRA DATA SEGMENTS/PROCESS = XXX.?	4		4	4	4	4	4	4	— —
STD STACK SIZE = XXXX.?	800		800	800	800	800	800	800	WORDS

CONFIGURATOR QUESTION / MEMORY SIZE IN KILOWORDS	UNIT OF MEASURE	MINIMUM ALLOWED	MAXIMUM ALLOWED
MAX # OF OPEN SPOOLFILES = XXX.?	— —	0	256-N[1]
SHAREABLE AREA CST = XXX.?	ENTRIES	2	192
PROGRAM AREA CST = XXXX.?	ENTRIES	1	8192
DST = XXXX.?	ENTRIES	1	1024
PCB = XXX.?	ENTRIES	2	256
I/O QUEUE = XXX.?	ENTRIES	20	255[2]
TERMINAL BUFFERS = XXX.?	BUFFERS	10	255[3]
SYSTEM BUFFERS = XXX.	BUFFERS	8	255
MEMORY MANAGEMENT TABLE = XXXX.?	ENTRIES	16	1024
ICS = XXXX.?	WORDS	128	1024
UCOP REQUEST QUEUE = XXX.?	ENTRIES	1	256
TIMER REQUEST LIST = XXX.?	— —	1	128
BREAKPOINT TABLE = XXX.?	ENTRIES.	1	255
# RINS MIN = 5, MAX = XXXX.?	— —	5	1024
# OF GLOBAL RINS USED = 0, MAX = XXXX.?	— —	0	1024
# OF SECONDS TO LOGON = XXX.?	SECONDS	10	600
MAX # OF CONCURRENT SESSIONS = XXX.?	— —	1	255
MAX # OF CONCURRENT RUNNING JOBS = XXX.?	— —	1	255
DEFAULT JOB CPU TIME QUANTUM LIMIT = XXXXX.?	— —	0	32767
LOG FILE RECORD SIZE (SECTORS) = X.?	SECTORS	1	8
LOG FILE SIZE (RECORDS) = XXXXX.?	— —	16	32767
VIRTUAL MEMORY = XXXXX.?	SECTORS	1024	32767
DIRECTORY USED = XXX, MIN = 384 MAX = XXXX.?	SECTORS	384	6000
MAX # SPOOLFILES KILOSECTORS = XXXXXX.?	KILOSECTORS	0	262143(%777777D)[4]
# SECTORS PER SPOOLFILE EXTENT = XXXX.?	SECTORS	128	1024[5]
TIME QUANTUM = XXXXX.?	MILLISECONDS	100	32767
TERMINAL PRIORITY = XXX.?	— —	150	**250**
CS PRIORITY LIMIT = XXX.?	— —	150	**250**
DS PRIORITY LIMIT = XXX.?	— —	150	**250**
MAX # OF CONCURRENT RUNNING PROGRAMS = XXX.?	PROGRAM	1	127
MAX CODE SEG SIZE = XXXXX.?	WORDS	1024	16384
MAX # OF CODE SEGMENTS/PROCESS = XXX.?	— —	0	**63**
MAX STACK SIZE = XXXXX.?	WORDS	256	31232
MAX EXTRA DATA SEG SIZE = XXXXX.?	WORDS	0	32767
MAX # EXTRA DATA SEGMENTS/PROCESS = XXX.?	— —	0	255
STD STACK SIZE = XXXX.?	WORDS	256	4096

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).

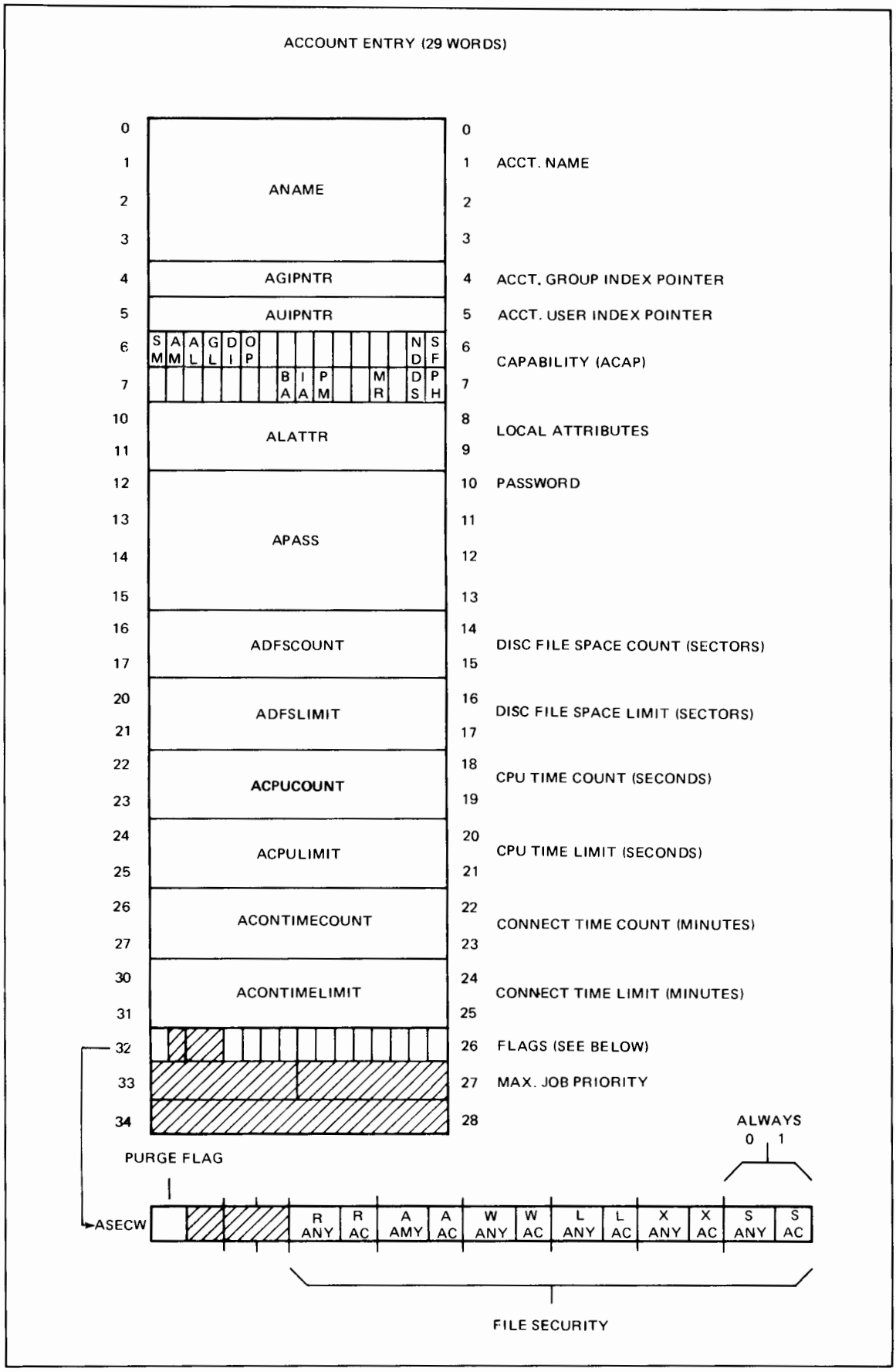


Figure D-1. Account Entry Format

```
:LISTACCT @

A=  A
040440 020040 020040 020040 001253 001252 177003 001761 A...............
000000 000000 040523 042106 020040 020040 000000 000000 ....ASDF........
077777 177777 000000 000143 077777 177777 000000 000040 .......C........
077777 177777 002525 000226 000000                      .....U....

A=  ADMIN
040504 046511 047040 020040 000114 000113 071003 000702 ADMIN....L.KR...
000000 000000 042501 051531 020040 020040 000000 103020 ....EASY........
077777 177777 000000 061523 077777 177777 000000 005661 ......CS........
077777 177777 002525 000036 000000                      .....U....

A=  DACOM
042101 041517 046440 020040 000460 000457 041003 000600 DACOM....0..B...
000000 000000 020040 020040 020040 020040 000000 000000 ................
077777 177777 000000 000000 077777 177777 000000 000000 ................
077777 177777 002525 000226 000000                      .....U....

A=  DCOM
042103 047515 020040 020040 000104 000103 071003 001773 DCOM.....D.CR...
000000 000000 042103 020040 020040 020040 000000 000000 ....DC..........
077777 177777 000000 000003 077777 177777 000000 000000 ................
077777 177777 002525 000226 000000                      .....U....

A=  LANG
046101 047107 020040 020040 000034 000033 071003 000701 LANG........R...
000000 000000 020040 020040 020040 020040 000000 060542 ..............AB
077777 177777 000205 031314 077777 177777 000000 071745 ......2.......S.
077777 177777 002525 000036 000000                      .....U....

A=  SYS
051531 051440 020040 020040 000007 000006 177003 001773 SYS.............
000000 000000 020040 020040 020040 020040 000000 054242 ..............X.
077777 177777 000205 000533 077777 177777 000000 027204 ................
077777 177777 004531 000036 000000                      .....Y....

A=  USERSLIB
052523 042522 051514 044502 000556 000555 071003 000713 USERSLIB.N.MR...
000000 000000 020040 020040 020040 020040 000000 000700 ................
077777 177777 000000 005326 077777 177777 000000 000503 ..............C
077777 177777 002525 000226 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. A sample listing is shown in figure D-4; in this example, the user wants to list all groups in the account named DIAG.

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 Intrinsic Reference Manual*).

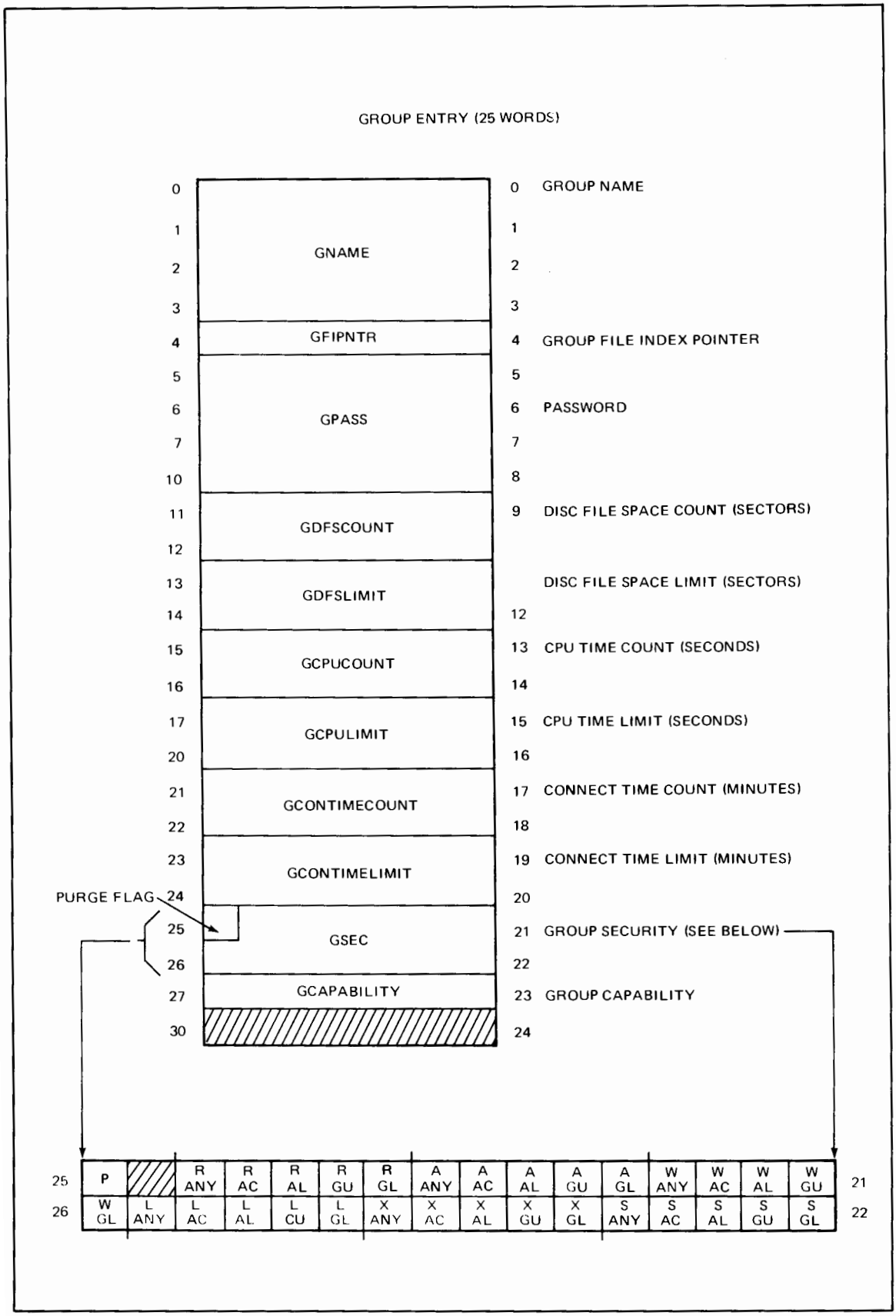


Figure D-3. Group Entry Format

```
:LISTGROUP @.DIAG

G=  CURTIS
041525 051124 044523 020040 000345 042115 041123 044124 CURTIS....NONONO
020040 000000 000000 077777 177777 000000 000003 077777 ................
177777 000000 000025 077777 177777 002041 004102 000600 .............B..
000000                                                  ..

G=  DELANO
042105 046101 047117 020040 000337 042104 020040 020040 DELANO....DD....
020040 000000 000000 077777 177777 000000 000000 077777 ................
177777 000000 000000 077777 177777 002041 004102 000600 .............B..
000000                                                  ..

G=  LEMOS
046105 046517 051440 020040 000335 052117 050440 020040 LEMOS.....TOQ...
020040 000000 000000 077777 177777 000000 000000 077777 ................
177777 000000 000000 077777 177777 002041 004102 000600 .............B..
000000                                                  ..

G=  MACH
046501 041510 020040 020040 000631 044101 047101 020040 MACH......HANA..
020040 000000 000000 077777 177777 000000 000001 077777 ................
177777 000000 000001 077777 177777 002041 004102 000600 .............B..
000000                                                  ..

G=  PUB
050125 041040 020040 020040 000075 020040 020040 020040 PUB.............
020040 000000 000000 077777 177777 000000 000003 077777 ................
177777 000000 000000 077777 177777 020143 015006 000700 ...........C....
000000                                                  ..

G=  RADVANY
051101 042126 040516 054440 000341 042515 051131 020040 RADVANY...EMRY..
020040 000000 000000 077777 177777 000000 000000 077777 ................
177777 000000 000000 077777 177777 002041 004102 000600 .............B..
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.

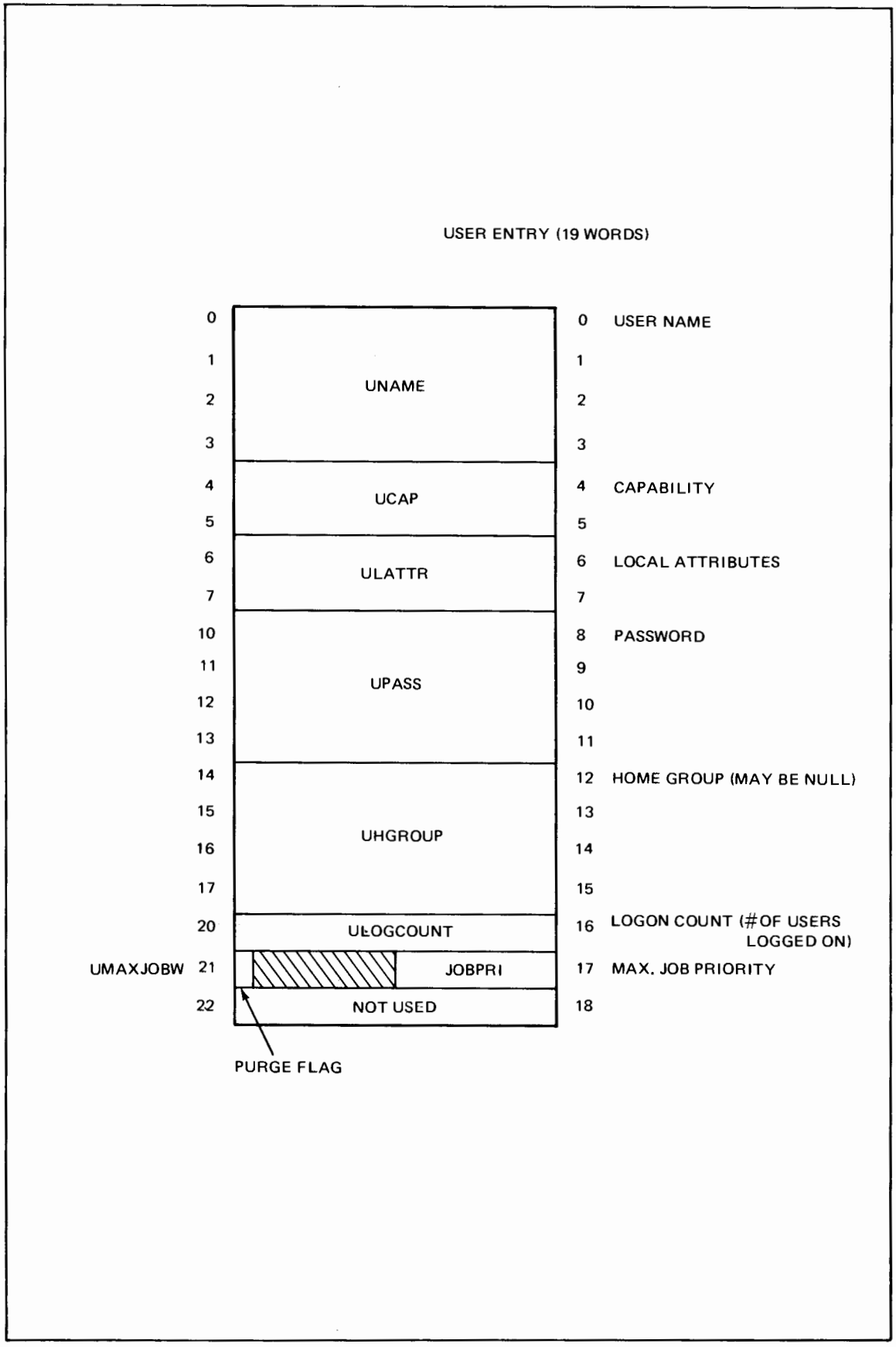


Figure D-5. User Entry Format

```
: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 three groupings, each headed by Q PIN JOBNUM.

Processes in the right-hand group are listed in high-to-low priority 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 second group, also listed in high-to-low priority, are willing to yield their main memory resources to other processes. Memory is taken automatically (segment by segment) from lower priority processes first.

Processes in the left-hand group are listed in numerical order and have no main memory resources. These processes are waiting on longer term events, and will again contend for the CPU when those events occur.

```
:SHOWQ

Q   PIN    JOBNUM        Q   PIN    JOBNUM        Q   PIN    JOBNUM

L     4                  L    13                 C   M21
L     6                  L     5
L    11                  L    10
L    14                  L    12
                         L     3
                         C   M20
                         L    16
                         L     7
                         L    15
                         L    17

15 PROCESSES; 500 QUANTUM, 152 TPRI, 160 CPRI, 180 DPRI
```

Figure D-7. :SHOWQ List Format

After displaying the process queues, =SHOWQ prints the number of processes and the scheduling parameters currently in effect.

NOTE

On occasion, a process will appear in the right-hand group as well as in one of the groups to the left. This indicates that the process was being moved to the right during the execution of =SHOWQ.

The following information is displayed for each process:

$$\left\{\begin{matrix}\mathrm{L}\\ \mathrm{C}\\ \mathrm{D}\\ \mathrm{E}\end{matrix}\right\} \left[\left\{\begin{matrix}\mathrm{M}\\ \mathrm{U}\end{matrix}\right\}\right] \quad \langle pin\rangle \left[\#\left\{\begin{matrix}\mathrm{J}\\ \mathrm{S}\end{matrix}\right\} \quad \langle jobnum\rangle\right]$$

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.

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 29 are reserved for the disc volume label, defective tracks table,and bootstrap input/output program. The disc free space table starts in Sector 30 and ranges in length between 12 and 32 sectors, depending on the type of disc.

Following the free space table are the disc directory and virtual memory. Their sizes are defined by parameters specified at system configuration time. (See below for the guidelines for determining the sizes of these two areas.)

The next portion of the disc is used for the system files, message catalog, and tables used by INITIAL and INITIAL program segments. At the present time, this accounts for approximately 2600 sectors. (This includes the SL file containing the Scientific, Compiler and COBOL/3000 Libraries.)

The disc directory currently has the following approximate maxima: 650 accounts; 155 groups per account; 200 users per account; and 1385 files per group. The following formula should be used to determine the approximate numbers of sectors needed for the directory:

$$\text{Sectors} = 6 + 6 * A + \lceil (4.2 * G) \rceil + \lceil (.15 * U) \rceil + \lceil (.05 * F) \rceil$$

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.

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." 14-22: Used by MPE. 23: Real 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.
4	EOF Mark
5	Tape directory — Consists of 12-word entries, blocked 85 per 1020 — word record. (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 with 1024 words per physical tape record. (The last record may be shorter, but will always be a multiple of 128 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.
8	EOF Mark
9	Second File

Table F-1. :STORE Tape Format (continued)

ITEM NO.	ITEM
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** 21 — =1 means that preceding file ended with preceding EOF mark 22 — =1 means that entire tape set ends with preceding EOF mark
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.

Table F-2. :SYSDUMP Tape Format

ITEM NO.	ITEM
1	Bootstrap — A record, 40 words long, read by the cold load microcode and containing a six-word SIO program which reads in the next record.
2	I/O program — An input/output program to read the records defined in Items 3 through 18.
3	ICS and entry point (34 words) — First 32 words are the Interrupt Control Stack (ICS) containing the initial values for DB, DL, Z, Q and S. Words 32-33 are the SETR and EXIT instructions which will be executed when the RUN switch is pressed.
4	Low-core (12 words). Initial values for absolute locations 0-11.
5	Disc Cold Load Information Table (128 words) — Sector 18 of the system disc.
6	Old Volume Table (280 words) — Volume table as it was before any changes made to it on this :SYSDUMP.
7	Volume Table (280 words) — Modified volume table.
8	Logical-Physical Device Table (512 words) — As updated by input/output configuration changes.
9	Logical Device Table (2 records of 640 words each) — As updated by input/output configuration changes.
10	Device Class Table (2 records of 768 words each) — As updated by input/output configuration changes.
11	Driver Table (2 records of 640 words each). Driver name for each DRT and core-resident flag, as updated by configuration changes.
12	Core Size Related Configuration Table (640 words) — 128 words for each of the 5 configurable core sizes.
13	Configuration Table (128 words) — Configuration information not related to core size.
14	Internal Interrupts (64 words) — Halt instruction for each of the internal interrupts.
15	Temporary CST (128 words) — Code segment table (CST) used for INITIAL's segments while INITIAL is running.
16	INITIAL's DB area (Blocked 1024 words per record) — Initialized DB area for INITIAL.
17	INITIAL marker (12 words) — Initial stack marker for INITIAL program.
18	INITIAL Segments 16-5 (Blocked 1024 words per record for each segment) — Those segments of INITIAL which are read in by the cold load input/output program.
19	INITIAL Segments 4-0 (Blocked 1024 words per record for each segment) — Those segments of INITIAL which are not in core when the machine halts following cold load.
20	Message catalog (Blocked 1024 words per record) — Table of MPE messages.
21	If any files are to be dumped by :SYSDUMP, this is the current RIN table, blocked 1024 words per record and the used portion of the disc directory, blocked 1024 words per record.

Table F-2. :SYSDUMP Tape Format (continued)

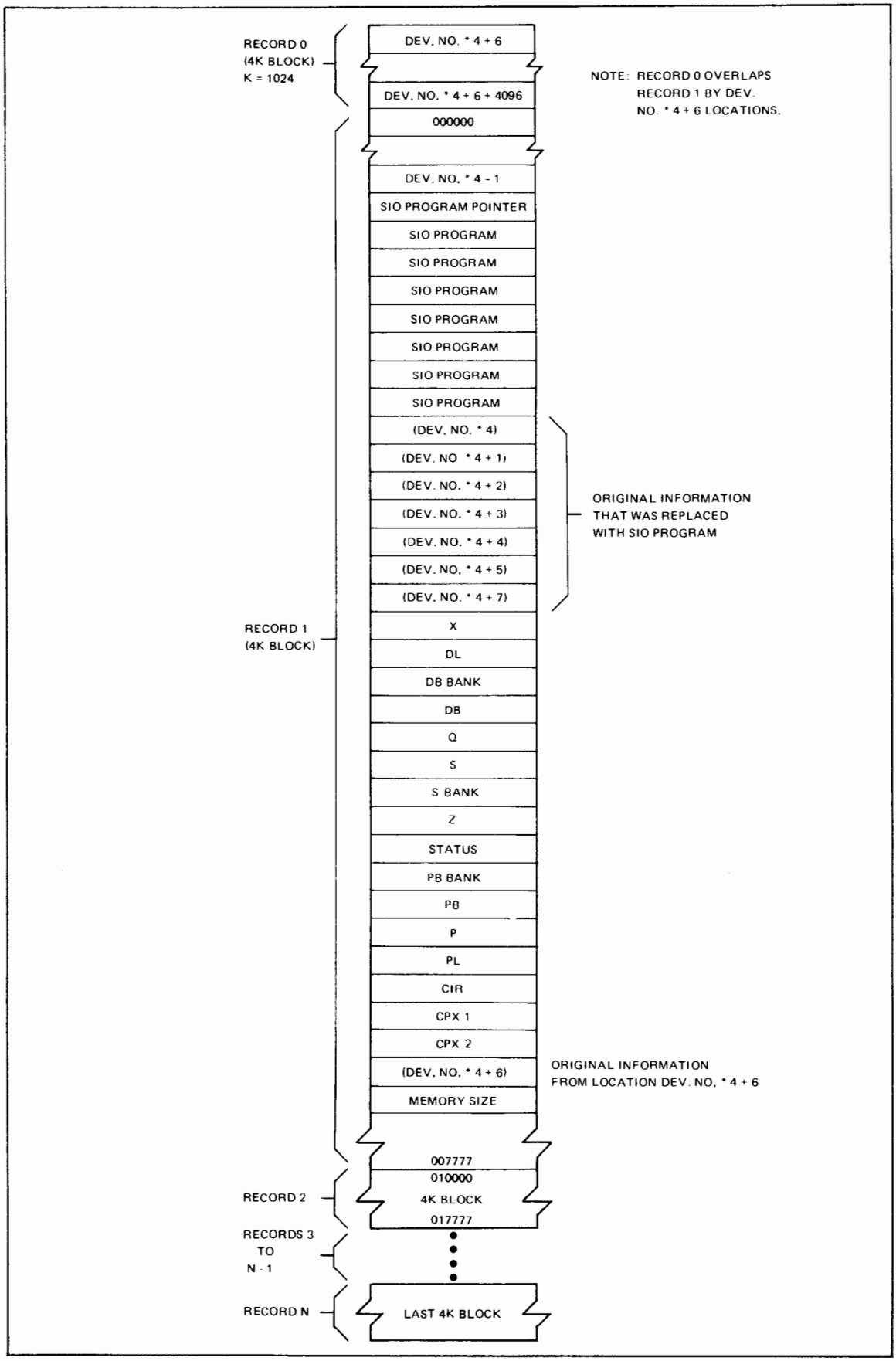
ITEM NO.	ITEM
22	End-of-File mark.
23	System files — Each allocated extent of each of the system files is dumped, blocked 1024 words per record. Files always dumped if they are on the disc (all in PUB.SYS) are: SL, SYSDUMP, INITIAL, CONFDATA, SEGPROC, SEGDVR, DISPATCH, LOAD, UCOP, PROGEN, DEVREC, ININ, LOG, IOTERM0, IOCDRD0, IOLPRT0, IOTAPE0, IOFDISC0, IOMDISC0, PFAIL, DEVDATA, IOMDISC1, IOPLOT0, IOPRPN0, IOPTPN0, IOPTRO0, IOREM0. In addition, any input/output drivers defined in the configuration but not in the above list are dumped.
24	End-of-File Mark.
25	If any user files were to be dumped, the rest of the tape is identical to the :STORE tape format described in Appendix N, (beginning with Item 3, the header label). The presence of the two file marks written as Items 22 and 24 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-3. 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	On the system disc, this field contains the bootstrap input/output program. On other discs, this field is filled with zeros.
6 (Bits 6:6) (Bits 12:4)	Disc type. Disc sub-type.
7	Cold-load count (incremented each time the system is cold-loaded).
8-9	The characters "3000," used to verify that the disc label is valid.
10-13	Volume name (Left-justified and padded with blanks).
14-127	Reserved (Initialized to zeros).

Table F-4. System Dump Magnetic Tape Format



MESSAGE CATALOG TABLE

APPENDIX G

MESSAGE CATALOG TABLE DESCRIPTION

The Message Catalog resides on system disc, and consists of one or more pages. A "page" is N sectors (where N is greater than 0) which are read one at a time as needed into an extra data segment.

Figure G-1 shows the general layout of the message catalog table, and Figure G-2 shows the entry format.

CATALOG CREATION

Each catalog entry must be in card image form (i.e., 80 characters per image), two cards per entry. The card 1 image contains everything except the message itself. Its format is free field, with blanks as delimiters. Data must be entered to the catalog file CATALOGS in the operating system group of the SUPPORT account as follows:

	PARAMETER #	PARAMETER
	4	REPLY TYPE (0, 1, 2 OR 3)
	2	MAX. REPLY STRING LENGTH
	3	I/O TYPE (0, 1 OR 2)
	4	PREFIX # (0, 1, 2, 3, 4 OR 5)
	5	FORMAT # (0, 1, 2 OR 3)
	6	I/O CONTROL PARAMETER
	7	RAW MESSAGE LENGTH (IN BYTES)
	8	# VARIABLES (4 MAX.)
VARIABLE #1	9	PARAMETER TYPE (0, 1, 2)
	10	PARAMETER POSITION (STARTS AT WHICH BYTE IN RAW MESSAGE - NUMBERING STARTS AT 0)
VARIABLE #2	11	
	12	
	ETC	

Card image 2 contains the raw message. Every character from column 1 to the end of the message is significant.

When all changes are complete, a :SYSDUMP with changes to the catalog file must be made to incorporate the new catalog in the system copy on magnetic tape. The new magnetic tape must then be used in a cold-load operation in order to replace the old catalog on disc.

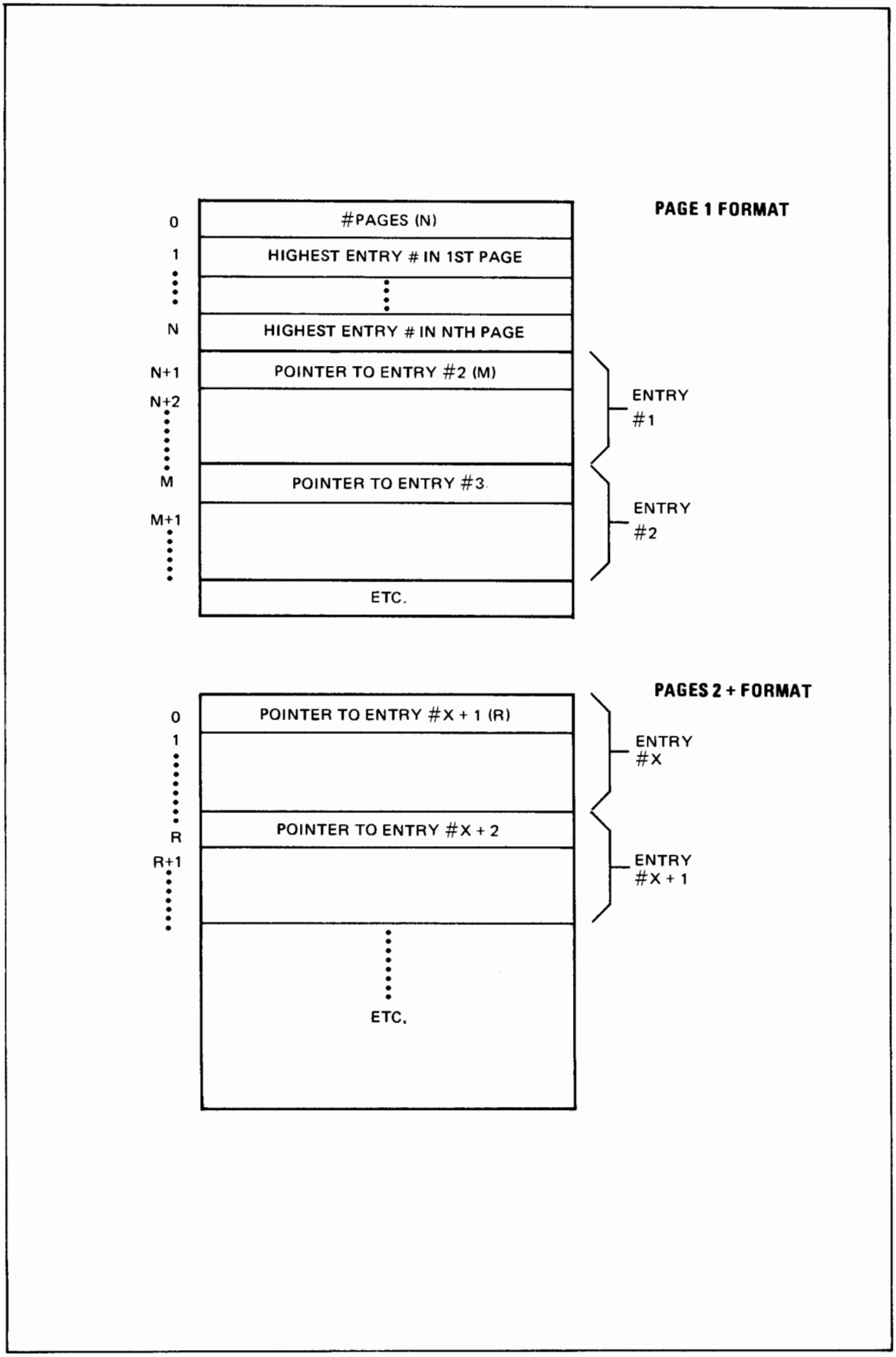


Figure G-1. Message Catalog General Layout

WORD =		CONTENT
	0 • (0:1)	REPLY FLAG (ON ➡ REPLY REQUIRED)
	0 • (1:2)	REPLY TYPE (0, 1, 2 OR 3) — ONLY FOR MESSAGES REQUIRING REPLIES, I.E. REPLY FLAG = 1
	0 • (3:5)	MAX. STRING LENGTH (REPLY TYPE 2 ONLY) — ONLY FOR MESSAGES REQUIRING REPLIES, I.E. REPLY FLAG = 1
	0 • (8:2)	I/O TYPE (0, 1 OR 2)
	0 •(10:3)	PREFIX # (0, 1, 2, 3, 4, 5, OR 6)
	0 •(13:3)	FORMAT # (ONLY FOR PREFIX NUMBERS 0, 1 OR 2)
	1	I/O CONTROL PARAMETER
	2 • (0:8)	MESSAGE LENGTH (WITHOUT PARAMETERS) IN BYTE
	2 • (8:8)	#PARAMETERS (4-MAX)
PARAMETER #1	3 • (0:2)	PARAMETER TYPE (0, 1 OR 2)
PARAMETER #1	3 •(2:14)	PARAMETER POSITION (STARTS AT WHICH BYTE IN RAW MESSAGE) NOTE: BYTE NUMBERING STARTS WITH ZERO
PARAMETER #2	4 • (0:2)	PARAMETER POSITION (STARTS AT WHICH BYTE IN RAW MESSAGE)
PARAMETER #2	4 •(2:14)	NOTE: BYTE NUMBERING STARTS WITH ZERO
	• • • • • • • • M • • • • N	MESSAGE

REPLY TYPE
- 0 – NUMBER (NUM)
- 1 – YES OR NO (Y/N)
- 2 – STRING (SXX) - (XX IS MAX. #CHAR ALLOWED IN REPLY)
- 3 – YES, NO OR NUMBER (YN#)

I/O TYPE
- 0 – STANDARD
- 1 – SOFT PRE-EMPTIVE
- 2 – HARD PRE-EMPTIVE

PARAMETER TYPE
- 0 – OCTAL NUMBER
- 1 – DECIMAL NUMBER
- 2 – STRING

PREFIX NUMBER
- 0 – ST
- 1 – IO
- 2 – MS

  CONSOLE
- 3 – ABORT
- 4 – FROM
- 5 – WARN

  USER
- 6 – NO PREFIX (ONLY THE RAW MESSAGE IS OUTPUT - USED BY COMMAND INTERPRETER FOR ERROR MESSAGES)

FORMAT NUMBER
- 0 – ? < PREFIX > / < TIME > / < PIN > / < MESSAGE > / < REPLY TYPE >
- 1 – Δ< PREFIX > / < TIME > / < MESSAGE >
- 2 – ? < PREFIX > / < TIME > / # { J, S, T } XXX/ < PIN > / < MESSAGE > < REPLY TYPE >
- 3 – Δ< PREFIX > / < TIME > / # { J, S, T } XXX/< MESSAGE >

Figure G-2. Message Catalog Entry Format

DISC PACK FORMATTING

APPENDIX H

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

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.00.0)
   >10
   ```

   You can now enter the dialogue necessary to format a disc pack.

5. 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 diagnostics 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
```

In all of the above examples, when a pack has been formatted, just continue typing in new programs or restart from step 1.

INDEX

Part No. 30000-90014
Printed in U.S.A. 2/77
Updates Nos. 1 and 2 Incorporated

HEWLETT hp PACKARD

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