

HP 3000 Computer Systems

MPE V/E Commands Reference Manual



HP Part No. 32033-90006
Printed in U.S.A. 1989

Second Edition
E1089

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

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

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

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

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

Print History

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

Edition	Date	Software Version
First Edition	February 1986	G.02.00
Update #1	August 1986	G.02.B0
Update #2	June 1987	G.03.00
Update #3	October 1988	G.03.04
Second Edition	October 1989	G.03.08

MPE V Manual Plan

MPE V Manual Plan

INTRODUCTORY LEVEL:

**GENERAL INFORMATION
Manual
(5953-7553)**

**MPE V/E HP 3000:
Fundamental Skills
(Part No. Pending)**

**HP 3000 GUIDE
FOR THE
NEW SYSTEM OPERATOR
(32033-90021)**

**MPE V/E HP 3000:
Advanced Skills
(Part No. Pending)**

**MPE V GENERAL USER'S
Reference Manual
(32033-90158)**

**FCOPY
Reference Manual
(32212-90003)**

STANDARD USER LEVEL:

**MPE V COMMANDS
Reference Manual
(32033-90006)**

**MPE V INTRINSICS
Reference Manual
(32033-90007)**

**MPE V SYSTEM UTILITIES
Reference Manual
(32033-90008)**

**MPE V SEGMENTER
Reference Manual
(30000-90011)**

**MPE V DEBUG/
STACK DUMP
Reference Manual
(30000-90012)**

**MPE V FILE SYSTEM
Reference Manual
(30000-90236)**

ADMINISTRATIVE LEVEL:

**MPE V SYSTEM
OPERATION AND
RESOURCE MANAGEMENT
Reference Manual
(32033-90005)**

**MPE V SECURITY AND
ACCOUNT STRUCTURE
(32033-90136)**

**MPE V STORING AND
RESTORING FILES
(32033-90133)**

**MPE V SYSTEM BACKUP
AND RECOVERY
(32033-90134)**

SUMMARY LEVEL:

**MPE V QUICK
REFERENCE GUIDE
(32033-90023)**

Preface

The second edition of the *MPE V/E Commands Reference Manual* places all the MPE V/E commands within easy reach in a single volume. Whether you are a system operator, a system manager, an account manager, or you are just getting used to your HP 3000, you will find the MPE V/E commands you need here, arranged alphabetically.

There are references to data communications in the commands that “bridge” into the NS/3000 AdvanceNet subsystem or into the DS/3000 subsystem (**FILE** and **RESET**). AdvanceNet and DS/3000 are not part of the MPE V/E fundamental operating system and must be purchased separately. The data communications parameters for **FILE** and **RESET** are given here in order to represent the command syntax fully and accurately. Omitting the parameters does not hinder your use of those commands. If you attempt to use those parameters and your system does not support data communications, you will get an error.



How to Use This Manual

This manual is divided into five sections.

Chapter 1 provides a general introduction to the MPE V/E commands and their structure. It also discusses some of the restrictions placed on their use. Read this for an overview of the system. For a more detailed introduction to MPE V/E, refer to the *MPE V General User's Reference Manual* (32033-90158).

Chapter 2 contains the reference specifications for all the MPE V/E commands available to MPE V/E users, presented alphabetically by command name. You can look up each command and find the following information:

- Command syntax.
- Parameter definitions, including meaning, constraints, and defaults.
- When and how the command may be used, such as in jobs, sessions, in **BREAK**, or programmatically.
- Whether the command can be interrupted with the **BREAK** key.
- The capability required to use it; for example, system supervisor (OP), system manager (SM), or account manager (AM), or whether the command must be executed at the console.
- A description of how to use the command, and any related special considerations.
- An example.
- Additional discussion, if any, in this or another manual.

Chapter 3 describes user-defined commands (UDCs) and how you can create your own.

Appendix A lists the terminals supported by MPE V/E.

Appendix B gives the formal file designator for each command parameter to which it applies.

Additional Reference

The following documentation may also be helpful:

- *MPE V Intrinsic Reference Manual* (32033-90007)
- *MPE File System Reference Manual* (30000-90236)
- *MPE V Utilities Reference Manual* (32033-90008)
- *MPE V Storing and Restoring Files* (32033-90133)
- *MPE V System Backup and Recovery* (32022-90134)
- *MPE V Account Structure and Security* (32033-90136)
- *MPE V System Operator and Resource Management Reference Manual*(32033-90005)
- *HP AdvanceNet NS/3000 User/Programmer Reference Manual* (32344-90001)

Conventions

UPPERCASE

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

COMMAND

can be entered as any of the following:

command Command COMMAND

It cannot, however, be entered as:

comm com_mand comamnd

italics

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

COMMAND *filename*

bold italics

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

COMMAND(*filename***)**

punctuation

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

(*filename*):(*filename*)

underlining

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

Do you want to continue? >> yes

{ }

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

**COMMAND { ON
 OFF }**

Conventions (continued)

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

COMMAND *filename* [OPTION]

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

COMMAND *filename* $\left[\begin{array}{l} \text{OPTION} \\ \text{parameter} \end{array} \right]$

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

[,*parameter*][...]

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

[*parameter*][, ...]

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

$\left\{ \begin{array}{l} \text{A} \\ \text{B} \end{array} \right\} | \dots |$

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

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

(*parameter*)Δ(*parameter*)

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

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

Conventions (continued)

base prefixes

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

%num specifies an octal number.

#num specifies a decimal number.

\$num specifies a hexadecimal number.

If no base is specified, decimal is assumed.

bits (*bit:length*)

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

```

most significant                                     least significant
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 |   |   |   |   |   |   |   |   |   |   |   |   |   | 13|14|15|
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
bits (0:1)                                           bits (13:3)

```


Contents

1. Introduction

Special Capabilities	1-2
Operator Commands	1-2
Moving the System Console	1-3
Command Elements	1-3
Prompt	1-4
Command Name	1-4
Parameters	1-4
Positional Parameters	1-4
Keyword Groups	1-4
Entering Commands	1-5
System-Defined Files	1-5
Continuation Characters	1-5
Sequence Numbers in Job Files	1-6
Command Errors	1-6
Executing Commands	1-6
Interrupting Command Execution	1-7
Aborting a Program	1-8
HELP Facility	1-8

2. Command Definitions

() COMMAND LOGON	2-8
ABORT	2-12
ABORTIO/=ABORTIO	2-13
ABORTJOB/=ABORTJOB	2-15
ACCEPT	2-17
ALLOCATE	2-19
ALLOW	2-21
ALTACCT	2-23
ALTGROUP	2-28
ALTJOB	2-32
ALTLOG	2-34
ALTSEC	2-36
ALTPOOLFILE	2-40
ALTUSER	2-42
ALTVSET	2-46
ASSOCIATE	2-48
AUTOALLOCATE	2-50
BASIC	2-51
BASICGO	2-53
BASICOMP	2-55
BASICPREP	2-57

BBASIC	2-59
BBASICGO	2-61
BBASICOMP	2-63
BBASICPREP	2-65
BREAKJOB	2-67
BUILD	2-68
BYE	2-75
CACHECONTROL	2-76
CHANGELOG	2-78
COBOL	2-80
COBOLGO	2-82
COBOLPREP	2-84
COBOLII	2-86
COBOLIIGO	2-88
COBOLIIPREP	2-90
COMMENT	2-92
CONSOLE	2-93
CONTINUE	2-95
DATA	2-96
DEALLOCATE	2-99
DEBUG	2-100
DELETESPOOLFILE	2-101
DISALLOW	2-103
DISASSOCIATE	2-105
DISCRPS	2-106
DISMOUNT	2-107
DOWN	2-108
DOWNLOAD	2-110
DSTAT	2-112
EDITOR	2-114
ELSE	2-115
ENDIF	2-116
:EOD	2-117
:EOF:	2-119
EOJ	2-120
FCOPY	2-121
FILE	2-123
FOREIGN	2-138
FORTGO	2-140
FORTPREP	2-142
FORTTRAN	2-144
FREERIN	2-146
FTN	2-147
FTNGO	2-149
FTNPREP	2-151
FULLBACKUP	2-153
GETLOG	2-154
GETRIN	2-156
GIVE	2-157
HEADOFF	2-158
HEADON	2-159

HELLO	2-160
HELP	2-166
IF	2-169
JOB	2-171
JOBFENCE	2-176
JOBPRI	2-178
JOBSECURITY	2-179
LDISMOUNT	2-180
LIMIT	2-181
LISTACCT	2-183
LISTEQ	2-186
LISTF	2-187
LISTFTEMP	2-195
LISTGROUP	2-203
LISTLOG	2-206
LISTUSER	2-208
LISTVS	2-211
LMOUNT	2-214
LOG	2-216
=LOGOFF	2-217
=LOGON	2-218
MOUNT	2-219
NEWACCT	2-221
NEWGROUP	2-225
NEWUSER	2-228
NEWVSET	2-231
OPENQ	2-233
OUTFENCE	2-234
PARTBACKUP	2-236
PASCAL	2-238
PASCALGO	2-240
PASCALPREP	2-242
PASSWORD	2-244
PREP	2-246
PREPRUN	2-249
PTAPE	2-254
PURGE	2-255
PURGEACCT	2-256
PURGEGROUP	2-258
PURGEUSER	2-260
PURGEVSET	2-261
RECALL/=RECALL	2-262
REDO	2-263
REFUSE	2-265
RELEASE	2-266
RELLOG	2-269
RENAME	2-270
REPLY/=REPLY	2-272
REPORT	2-274
RESET	2-277
RESETACCT	2-278

RESETDUMP	2-279
RESTORE	2-280
RESUME	2-288
RESUMEJOB	2-289
RESUMELOG	2-290
RESUMESPOOL	2-291
RPG	2-293
RPGGO	2-295
RPGPREP	2-297
RUN	2-300
SAVE	2-304
SECURE	2-306
SEGMENTER	2-307
SET	2-308
SETCATALOG	2-309
SETDUMP	2-312
SETJCW	2-314
SETMSG	2-319
SHOWALLOCATE	2-320
SHOWALLOW	2-323
SHOWCACHE	2-324
SHOWCATALOG	2-326
SHOWCOM	2-328
SHOWDEV	2-330
SHOWIN	2-333
SHOWJCW	2-337
SHOWJOB	2-339
SHOWLOG	2-344
SHOWLOGSTATUS	2-345
SHOWME	2-347
SHOWOUT	2-349
SHOWQ	2-354
SHOWTIME	2-356
=SHUTDOWN	2-357
SHUTQ	2-359
SPEED	2-360
SPL	2-362
SPLGO	2-364
SPLPREP	2-366
STARTCACHE	2-369
STARTSESS	2-370
STARTSPOOL	2-374
STOPCACHE	2-376
STOPSPPOOL	2-377
STORE	2-379
STREAM	2-385
STREAMS	2-391
SUSPENDSPOOL	2-393
SWITCHLOG	2-394
SYSDUMP	2-395
TAKE	2-397

TELL	2-398
TELLOP	2-400
TUNE	2-401
UP	2-404
VINIT	2-405
VMOUNT	2-408
VSUSER	2-410
WARN	2-411
WELCOME	2-413

3. User-Defined Commands

Syntax of UDCs	3-2
Header	3-2
The Command Name	3-2
Parameters	3-2
Execution Options	3-3
Body	3-6
Using the SETCATALOG Command	3-6
Building and Modifying a UDC File	3-8
Details of UDC Operation	3-11
Nesting UDCs	3-12
Errors in a UDC	3-13
Using Parameters	3-15
Types of Parameters	3-15
Parameter Delimiters	3-16
Increasing UDC Functionality	3-18
Using UDCs with Job Control Words	3-18

A. Terminals Supported by MPE V

B. Subsystem Formal File Designators

Index

Section Divider

1. Introduction



Introduction

MPE V/E commands allow you to start, control, and stop the processing of programs and to request various other system operations. You generally use them to perform functions external to the source-language programs that you write, although many of the commands and their functions may support your programs. For example, you use commands to:

- Initiate an interactive session (**HELLO**) or batch job (**JOB**)
- Display status information about sessions and jobs in the system (**SHOWJOB**)
- Create, save, and delete files (**BUILD**, **SAVE**, and **PURGE**, respectively)
- Specify and list file characteristics (**FILE** and **LISTF**)
- Dump files to tape or serial disk and subsequently restore them to the system (**STORE** and **RESTORE**), and specify security provisions for them (**ALTSEC**, **RELEASE** and **SECURE**)
- Compile programs (**FORTTRAN**, **FTN77**, **COBOL**, **COBOLII**, **PASCAL**, **RPG**, **BBASIC**, **SPL**), prepare those programs (**PREP**), and execute them (**RUN**)
- Determine the status of devices (**SHOWDEV** and **DSTAT**)
- Determine the status of device files, which are disk files originating on or destined for nonsharable devices (**SHOWIN** and **SHOWOUT**)
- Communicate with other users (**TELL**) and with the system operator (**TELLOP**)
- Define your own commands (**SETCATALOG**)
- Access private disk volumes (**MOUNT** and **DISMOUNT**)
- Obtain assistance in using the command interpreter (**HELP**)

Chapter 2 contains a complete catalog of MPE V/E commands and their specifications.

Special Capabilities

You require special capabilities to execute some MPE V/E commands. Those you encounter most often are:

OP	System supervisor. A user whose account and user name have been assigned OP capability. The system supervisor is responsible for optimizing the performance of the system, and may execute commands like TUNE and ALLOCATE or log on with ;HIPRI . OP commands should not be confused with operator commands, which are explained below.
SM	System manager. A user whose account and user name have been assigned SM capability. The system manager creates the accounting structure and assigns appropriate capabilities and initial security provisions to each account. The system manager also works with the Hewlett-Packard systems engineer (SE) and the system supervisor to develop the MPE V/E configuration which best suits the needs of the users at each installation.
AM	Account manager. A user, selected by the system manager, whose user name has been assigned AM capability. The account manager establishes groups and users within a particular account and assigns capabilities and attributes to them.
CV	Create volumes capability. A capability assigned to accounts and user names that allows the user to create private volumes and access them.
PS	Programmatic sessions. A user who is allowed to execute the STARTSESS command and to call the STARTSESS and ABORTSESS intrinsics. This user is designated by the system manager during the creation of SYSTART.PUB.SYS for the startup state configurator, or by an applications programmer who creates a turnkey system.
NM	Node manager. A user, selected by the system manager, who is assigned to manage the data communications subsystem at a specified location.
NA	Network administrator. A user, selected by the system manager, who is assigned the capability to perform comprehensive diagnostics on an entire communications network.

Note



Execution of the commands in this manual does not require NM or NA capability. They are included to present a complete listing of the user capability list. Refer to the *HP Advance Net NS/3000 User/Programmer Reference Manual* (32344-90001).

Operator Commands

Commands that may be executed only from the console are operator commands. They do not require special capabilities, but they are usually executed only by the system operator, who is logged on at the system console. With five exceptions, even operator commands may be made available to other users working at standard terminals. They become available only if the operator specifically assigns (or “distributes”) them to particular users. The exceptions are **=ABORTIO**, **=ABORTJOB**, **=LOGON**, **=LOGOFF**, and **=SHUTDOWN** **RECALL**, **REPLY?**. These five remain under the exclusive jurisdiction of the system console. They are considered console commands,

since the system console is the only device that accepts **Ctrl** A and displays an “=” prompt in response.

Operator commands may be distributed to specific users, at the operator’s discretion, in two ways:

- With the **ALLOW** command
- With the **ASSOCIATE** command

The **ALLOW** command gives users the ability to execute specific operator commands at their own terminals. Any command usually executed at the system console may be allowed, except those which can be entered only at the “=” prompt.

The **ASSOCIATE** command is more limited than the **ALLOW** command. When a user is associated with a specific device class, he or she is granted operator control of any device belonging to that class. Before a user can be “associated”, the user name must be entered into a device/class user association table. This table is created by the system manager with the **ASOCTABL.PUB.SYS** utility or the **ASOCTABL5.PUB.SYS** utility.

Moving the System Console

Operator responsibilities can be reassigned to another user by transferring system console capabilities to another logical device, or *ldev*. This is called “moving the system console”. Because the system console is a logical designation rather than a physical device, moving it requires only that the system console be assigned, using the **CONSOLE** command, to a different *ldev*. Moving the system console requires **OP** capability since the console and its capabilities are privileged resources. The system console is usually configured as *ldev* 20 and, under normal circumstances, should not be moved to another logical device.

Command Elements

Each MPE V/E command consists of the following parts:

- A colon prompt (:), supplied by the system during interactive sessions; the user must provide the prompt, usually an exclamation point (!) or another substitute, for batch jobs.
- A command name (required in all cases).
- A parameter list (used in most cases).

Note



Commands may contain a maximum of 278 characters exclusive of prompting colons and continuation ampersands. MPE V/E does not permit a command name, keyword, positional parameter, or keyword subparameter to span more than one line. MPE V/E does not begin interpretation of a command until the last record of the command is read.

Prompt

The prompt identifies a statement as an MPE V/E command. In an interactive session, MPE V/E displays the colon at the terminal whenever it is ready to accept a command. You respond by entering the remainder of the command after the colon. In a batch job, however, you may enter an exclamation point (!) or other characters to simulate the colon prompt, placing it in column 1 and following it with the rest of the command.

Command Name

The command name, which you enter following the colon, requests a specific operation. MPE V/E prohibits embedded blanks within the name, and rejects the command if they appear. MPE V/E interprets the next nonalphabetic character encountered as the end of the command name; typically this character is a blank. Blanks may appear between the colon and the command name except in the HELLO, JOB, and DATA commands.

Parameters

The parameter list contains one or more options for the command. Parameters are required in some commands, but are optional or prohibited in others. Parameter lists may include positional parameters, keyword parameter groups, or both, separated from each other by delimiters such as commas, semicolons, "equals" signs, or other punctuation marks. MPE V/E permits both decimal and octal numbers as command parameters. Octal numbers are distinguished by their preceding percent sign (%).

If a parameter is required, a value for it must be included in the parameter list. MPE V/E substitutes default values for optional parameters that are omitted from the list. These default values are specified in the parameter definitions that accompany each command reference.

The end of a command is indicated by the last nonblank character of the record in which it appears. For terminal input, a carriage return usually signals termination of command input. However, if the last nonblank character of the record is a continuation character, the command input is continued to the next record.

Positional Parameters

Positional parameters are separated from one another by commas. When you omit an optional positional parameter from within a list, you must include the comma that would normally follow that parameter. When you omit an optional positional parameter immediately following a command name, a comma becomes the first character in the parameter list. When you omit positional parameters from the end of the list, you need not include commas to signify this; the terminating **Return** or end-of-record is sufficient.

Keyword Groups

You can enter keyword groups in any order with respect to each other. Each keyword group is preceded by a semicolon (;). When more than one subparameter appears in a group, they are usually separated from each other by commas. All delimiters can be preceded or followed by blanks. The following example shows both positional and keyword parameters. INPT and OUTP are the positional parameters. DL and CAP are keywords. PH, DS, and MR are subparameters of the CAP keyword:

```
:PREP INPT,OUTP;DL=500;CAP=PH,DS,MR
```

When both keyword groups and positional parameters form a list, positional parameters that are not associated with a keyword always precede the keyword groups. When you omit trailing parameters from the positional group, you need not include their delimiters. The occurrence of the first keyword signifies the termination of the positional group. When you omit optional subparameters from a keyword group, follow the same rules that apply to positional parameters.

Entering Commands

You can enter commands through any standard input device, usually a terminal (for sessions) or a stream file (for jobs). Each command is accepted by the MPE V/E command interpreter (CI), which passes it to the appropriate system procedure for execution. Following execution, control returns to the command interpreter, which is now ready for another command.

Note



After a command has been executed, the command interpreter prompts the input device for the next command. If this READ fails due to an I/O error, the CI tries to post another READ. If four consecutive READ attempts fail, the CI logs the user or job off the system.

System-Defined Files

All jobs and sessions have two system-defined files, `$STDIN` and `$STDLIST`, associated with them. `$STDIN` is a file name indicating the standard input file from which the job or session is initiated. Commands and data are read from `$STDIN` by the command interpreter. `$STDLIST` is a file name indicating the standard listing or output file corresponding to the particular job or session. Command output, error messages, etc. are written to `$STDLIST`. For sessions, both `$STDIN` and `$STDLIST` are typically terminal device files. For jobs, `$STDIN` is often a stream file or a card reader; `$STDLIST` is often a spool file destined for a line printer. For more information, refer to the `STREAM` command in chapter 2.

Continuation Characters

When the length of a command exceeds one record (for instance, one entry line on your terminal, or one source card), you may enter an ampersand (&) as the last nonblank character of the record and continue the command on the next record. The next record must begin with a colon, which is supplied automatically by MPE V/E in interactive processing. In batch processing, you must enter a colon or a special character to simulate the colon. You may embed blanks between the colon that begins the continuation record and the first nonblank character in the continuation record. In the following example, the command contains a continuation character at the end of the first line and an embedded blank at the beginning of the second:

```
:RUN PROGB;NOPRIV;LMAP;STACK=501;PARM=5;&  
: DL=600;LIB=G
```

Sequence Numbers in Job Files

Spooled job files may have sequence line numbers. MPE V/E determines whether a job file is sequenced or not by scanning the *JOB record. If the last eight columns of the JOB record are numeric, the job file is considered to be sequenced. MPE V/E ignore the last eight columns of each record of every file associated with that particular job.

If you use a sequenced job file, you must place ampersands (&) used as continuation characters before the sequence fields or they will not be read. Any programs that utilize sequence numbers for any purpose (for example, continuity check when doing a serial data read) do not function properly, as the sequence numbers are not read. A file used as input to an editor must have no data in the sequence fields as these last eight bytes are not read.

Command Errors

If you make an error while entering a command in an interactive session, MPE V/E suppresses execution of the command and attempts to determine the cause of the error. MPE V/E prints a caret (^) under the incorrect part of the command, along with the appropriate message, and control is returned to your terminal.

If you enter an incorrect command in a job file and the preceding command is not CONTINUE, MPE V/E flushes the job. MPE V/E prints an error message on the job's standard list device (usually the printer) so that you know which command caused the job to abort. However, if the erroneous command is preceded by the CONTINUE command, MPE V/E reports the error, and continues processing the job. For further information, refer to the JOB, :EOJ, and CONTINUE commands in chapter 2.

Executing Commands

MPE V/E allows you to execute nonprogram commands from within many subsystems. You do this from the subsystem prompt by preceding the MPE V/E command with a colon (:). Refer to table 1-1 for a list of nonprogram commands.

Many commands may be invoked during the execution of another operation, provided that the operation itself is "breakable." You may suspend the execution of the current operation by pressing the **BREAK** key on the terminal or by calling the CAUSEBREAK intrinsic.

Commands that may be issued from a program are available through the COMMAND intrinsic. This intrinsic invokes the command interpreter and passes the MPE V/E command for execution. The CAUSEBREAK and COMMAND intrinsics are explained in the *MPE V Intrinsics Reference Manual* (32033-90007).

Interrupting Command Execution

You can interrupt a command by pressing **BREAK** on your terminal. The effect that pressing **BREAK** has on your executing command depends on whether the command is a program command, or not.

Table 1-1 lists breakable and nonbreakable nonprogram commands. Table 1-2 lists breakable program commands.

Table 1-1. Nonprogram Commands

Nonbreakable			Breakable
() COMMAND LOGON	EOD	RELLOG	ALTSEC
ABORT	EOJ	RENAME	BYE
ABORTIO/=ABORTIO	FILE	REPLY/=REPLY	HELLO (If there is no OPTION LOGON UDC*)
ABORTJOB/=ABORTJOB	FOREIGN	RESET	HELP
ACCEPT	FREERIN	RESETACCT	LISTACCT
ALLOCATE	GETLOG	RESETDUMP	LISTF
ALLOW	GETRIN	RESETDUMP	LISTGROUP
ALTACCT	GIVE	RESUME	LISTLOG
ALTGROUP	HEADOFF	RESUMEJOB	LISTUSER
ALTJOB	HEADON	RESUMELOG	LISTVS
ALTLOG	IF	RESUMESPOOL	REDO
ALTSEC	JOB	SAVE	REPORT
ALTSPoolFILE	JOBFENCE	SECURE	RESTORE
ALTUSER	JOBPRI	SETCATALOG	RUN
ALTVEST	JOBSECURITY	SETDUMP	SEGMENTER
ASSOCIATE	LDISMOUNT	SETJCW	SHOWALLOCATE
AUTOALLOCATE	LIMIT	SETMSG	SHOWALLOW
BUILD	LMOUNT	SHOWCOM	SHOWCATALOG
CHANGELOG	LOG	SHOWLOG	SHOWDEV
COMMENT	=LOGOFF	SHOWLOGSTATUS	SHOWIN
CONSOLE	=LOGON	SHOWTIME	SHOWJCW
CONTINUE	MOUNT	=SHUTDOWN	SHOWJOB
DATA	NEWACCT	SPEED	SHOWME
DEALLOCATE	NEWGROUP	STARTSPOOL	SHOWOUT
DEBUG	NEWUSER	STOPSPool	SHOWQ
DELETESPOOLFILE	NEWVSET	STREAMS	STREAM
DISALLOW	OUTFENCE	SUSPENDSPOOL	SYSDUMP
SET	PASSWORD	SWITCHLOG	TUNE
DEBUG	PTAPE	TAKE	VINIT
DISASSOCIATE	PURGE	TELL	
DISMOUNT	PURGEACCT	TELLOP	
DOWN	PURGEGROUP	UP	
DOWNLOAD	PURGEUSER	VMOUNT	
DSTAT	PURGEVSET	VSUSER	
ELSE	RECALL/=RECALL	WARN	
ENDIF	REFUSE	WELCOME	
	RELEASE		

Table 1-2. Program Commands (All Are Breakable)

BASIC	COBOLII	PASCALGO
BASICGO	COBOLIIGO	PASCALPREP
BASICOMP	COBOLIIPREP	PREP
BASICPREP	COBOLPREP	PREPRUN
BBASIC	EDITOR	RPG
BBASICGO	FCOPY	RPGGO
BBASICOMP	FORTGO	RPGPREP
BBASICPREP	FORTPREP	SPL
COBOL	FORTRAN	SPLGO
COBOLGO	PASCAL	SPLPREP
		STORE

Note



User programs are initiated with the **BREAK** facility enabled. You may programmatically disable the **BREAK** facility by using the **FCONTROL** intrinsic. Application programs can restrict user access to the command interpreter by disabling the **BREAK** function. For more information, refer to the *MPE V Intrinsic Reference Manual* (32033-90007).

Aborting a Program

When a program contains logical errors, such as an infinite loop, you can abort it by following the steps below:

1. Press **BREAK**. This suspends your program and MPE V/E prompts you for a new command.
2. At the colon prompt, enter

:ABORT

This terminates the program, but does not disrupt the session.

MPE V/E confirms the termination of the program by displaying the following message and prompting you for a new command:

```
PROGRAM ABORTED PER USER REQUEST (CIERR 989)
:
```

The commands **BREAK**/ABORT and **RESUME** are valid only in sessions, not in batch jobs.

HELP Facility

MPE V/E also provides an online **HELP** facility. To access the **HELP** subsystem, enter:

:HELP

or **HELP** followed by the name of the command you want information about. For more details, refer to the **HELP** command in chapter 2.

Section Divider

2. Command Definitions



Command Definitions

This chapter provides information on all MPE V/E commands available to users. For your convenience they are arranged in alphabetical order. Each command specification contains the following information:

Command Name	The command name is listed at the beginning of each command description, followed by a brief definition of its function. Each command description starts on a new page.
Syntax	The syntax shows the exact form in which you enter the command and its parameters.
Parameters	Each parameter is defined and its function, limitations, and defaults are explained.
Use	Describes the conditions under which the command can be used (such as during a session, job, program, or in BREAK), also indicates whether the command can be interrupted with the BREAK key. If appropriate, special capabilities required to use it are listed.
Operation	Explains how the command operates and notes any special considerations.
Examples	Provides examples of how to use the command. The data that you as the user would enter after the colon prompt (:) is shown underlined. The system responses are shown in the same font, but not underlined. If you are required to enter a prompt, for example in a subsystem, the prompt itself is shown underlined.
Additional Discussion	Provides pointers to other commands or manuals that contain related information.

Some commands require special capabilities such as system supervisor (OP) or system manager (SM) for their use. These capabilities are listed in this chapter under ALTACCT. For more information, refer to the *MPE V General Users Reference Manual* (32033-90158). Table 2-1 summarizes the MPE V/E commands and their functions. It also lists any special capabilities a command may require. If there are none, the command is available to any user at any level. In general, the commands are set out exactly as a user would enter them after the CI colon prompt (:). For example, you would enter ABORT at the prompt as follows:

: ABORT

However, the “Control-A” commands are shown with their “=” prompt, even though the prompt is system-generated. These are the commands that you enter after you have executed **CTRL A** and the system displays the “=” prompt. For example, the “Control-A” version of **ABORTIO** is shown as “=**ABORTIO**” in table 2-1, although you would only enter **ABORTIO** after the “=” prompt, as follows:

: **CTRL A**

= **ABORTIO**

Several commands can be used directly or after **CTRL A**, such as **REPLY/=REPLY**.

Table 2-1. Functions of MPE V/E Commands

Function	Command	Capabilities Required (USER Unless Specified)
Running Sessions	() COMMAND LOGON ABORT BYE :EOF: :EOD HELLO HELP RESUME	
Running Jobs	COMMENT CONTINUE ELSE ENDIF EOJ IF JOB SET SETJCW SHOWJCW STREAM	
Determining Job or Session Status and Resources	REPORT SHOWJOB SHOWME	AM or SM for ACCT/SYS information
Job/Session Commands	=ABORTJOB =LOGOFF =LOGON =SHUTDOWN	All may be executed only from the console.
Job/Session Commands	ABORTJOB**	AM, SM, or from the console.
Job/Session Commands	ACCEPT ALLOW ALTJOB** BREAKJOB** DISALLOW JOBFENCE JOBSECURITY LIMIT LOG REFUSE RESUMEJOB**	All may be executed only from the console, unless distributed to other terminals. OP needed.
Job/Session Commands	STARTSESS	PS capability

** Higher job security or AM capability required to execute command on another job session.

Table 2-1. Functions of MPE V/E Commands (contd.)

Function	Command	Capabilities Required (USER Unless Specified)
Managing Files	ALTSEC ALTVSET BUILD DATA DISMOUNT FILE LISTEQ LISTF LISTFTEMP LISTVS MOUNT NEWVSET PURGE PURGEVSET RELEASE RENAME RESET RESTORE SAVE SECURE STORE VSUSER	CV UV or CV CV CV SM, OP or PM for privileged files. SM, OP or PM for privileged files.
Running Subsystems and User Programs Commands marked with an asterisk (*) are discussed in detail in separate subsystem manuals.	BASIC* BASICGO* BASICCOMP* BASICPREP* BBASIC* BBASICGO* BBASICCOMP* BBASICPREP* COBOL* COBOLGO* COBOLII* COBOLIIGO* COBOLIIPREP* COBOLPREP* DEBUG* EDITOR* FCOPY* FORTGO* FORTPREP* FORTRAN* FTN* (FORTRAN 77) FTNGO* FTNPREP* PASCAL* PASCALGO* PASCALPREP* PREP PREPRUN	PM

Table 2-1. Functions of MPE V/E Commands (contd.)

Function	Command	Capabilities Required (USER Unless Specified)
Running Subsystems and User Programs Commands marked with an asterisk (*) are discussed in detail in separate subsystem manuals.	RESETDUMP* RPG* RPGGO* RPGPREP* RUN SEGMENTER* SETDUMP* SPL* SPLGO* SPLPREP*	
Device and Device File Status	ASSOCIATE CONSOLE DISASSOCIATE DSTAT RECALL SHOWALLOW SHOWCACHE SHOWDEV SHOWIN SHOWOUT	AM or SM to access other accounts or groups.
Device and Device File Commands	=ABORTIO	May be executed only from the console.
Device and Device File Commands	ABORTIO ALTSPoolFILE CONSOLE DELETESPoolFILE DOWN DOWNLOAD FOREIGN GIVE HEADOFF HEADON LDISMOUNT LMOUNT OPENQ OUTFENCE RESUMESPool SHOWCOM SHUTQ STARTSPool STOPSPool STREAMS SUSPENDSPool TAKE UP VMOUNT	All may be executed only from the console, unless distributed to other terminals.

Table 2-1. Functions of MPE V/E Commands (contd.)

Function	Command	Capabilities Required (USER Unless Specified)
Requesting Utility Operations	FREERIN GETRIN PTAPE REDO SETCATALOG SETMSG SHOWCATALOG SHOWJCW SHOWTIME SPEED TELL TELLOP	
Resource Management	ALTLOG CHANGELOG GETLOG LISTLOG RELLOG SHOWLOGSTATUS	OP or LG OP or LG OP or LG OP or LG OP or LG
Creating Private Volumes	VINIT	SM or OP
Message Commands	REPLY/=REPLY WARN WELCOME	May be executed only from the console, unless distributed to other terminals.
Program/Procedure Allocation	ALLOCATE AUTOALLOCATE DEALLOCATE SHOWALLOCATE	OP or SM OP or SM OP or SM OP or SM
System Modification Backup	SYSDUMP	OP or SM
System Backup	PARTBACKUP FULLBACKUP	May be executed only from the console. OP needed.
Job and System Performance Management	CACHECONTROL DISCRPS JOBPRI SHOWCACHE STARTCACHE STOPCACHE TUNE	OP or SM Only from the console OP or SM OP or SM OP or SM OP or SM

Table 2-1. Functions of MPE V/E Commands (contd.)

Function	Command	Capabilities Required (USER Unless Specified)
System Reports and Information Management	RESUMELOG	OP or from the console.
	SHOWLOG	OP or from the console
	SHOWQ	OP or from the console
	SWITCHLOG	OP or from the console
Account Management	ALTACCT	SM
	ALTGROUP	AM
	ALTUSER	AM
	LISTACCT	AM or SM
	LISTGROUP	AM or SM
	LISTUSER	AM or SM
	NEWACCT	SM
	NEWGROUP	AM
	NEWUSER	AM
	PASSWORD	
	PURGEACCT	SM
	PURGEGROUP	AM
	PURGEUSER	AM
	REPORT	AM or SM
	RESETACCT	SM

() COMMAND LOGON

Begins a session, executes the MPE V/E command enclosed in parentheses, and ends the session upon completion of the command.

Syntax

$$\begin{aligned} & ([:] \textit{commandname}) [\textit{sessionname},] \textit{username} [/ \textit{userpass}] . \textit{acctname} [/ \textit{acctpass}] \\ & [, \textit{groupname} [/ \textit{grouppass}]] [; \textit{TERM} = \textit{termtype}] [; \textit{TIME} = \textit{cpusecs}] \left[; \textit{PRI} = \left\{ \begin{array}{l} \textit{BS} \\ \textit{CS} \\ \textit{DS} \\ \textit{ES} \end{array} \right\} \right] \\ & \left[\left\{ \begin{array}{l} ; \textit{INPRI} = \textit{inputpriority} \\ ; \textit{HIPRI} \end{array} \right\} \right] \end{aligned}$$

Parameters

<i>commandname</i>	Any MPE V/E or user-defined command (UDC), including its parameter list. The prompting colon may be omitted.
<i>sessionname</i>	An arbitrary name used in conjunction with <i>username</i> and <i>acctname</i> to form a fully qualified session identity. This name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The default is that no session name is assigned.
<i>username</i>	The user name, established by the account manager, that allows you to logon into this account. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character.
<i>userpass</i>	The user password, optionally assigned by the account manager. This password must contain from one to eight alphanumeric characters, beginning with an alphabetic character. A slash (/) must precede the <i>userpass</i> parameter.
<i>acctname</i>	The account name as established by the system manager. This name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. A period (.) must precede the <i>acctname</i> parameter.
<i>acctpass</i>	The account password, optionally assigned by the account manager. This password must contain from one to eight alphanumeric characters, beginning with an alphabetic character. A slash (/) must precede the <i>acctpass</i> parameter.
<i>groupname</i>	The group name to be used for the local file domain and the CPU time charges, as established by the account manager. It must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The default is your home group, if one is assigned. The parameter is required if a home group has not been assigned.

() COMMAND LOGON

<i>grouppass</i>	The group password, optionally assigned by the account manager. This password must contain from one to eight alphanumeric characters, beginning with an alphabetic character. You do not need to enter the <i>grouppass</i> parameter when you logon to your home group. However, if you request a logon group that has been assigned a password, you must supply the password when you logon to that group. A slash (/) must precede the <i>grouppass</i> parameter.
<i>termtype</i>	Defines the terminal type. The <i>termtype</i> parameter is the name of the file containing the desired characteristics for the terminal. The default value for <i>termtype</i>, a number from 0 to 22, is assigned by the system supervisor during system configuration. If your terminal is not of the default <i>termtype</i>, you must supply this parameter to ensure correct terminal input and output. The <i>termtype</i> file may not have a lockword or reside on a private volume. The workstation configurator utility allows you to create your own <i>termtype</i> files. You may specify the following characteristics of the terminal: data flow control, block mode, read trigger, special characteristics, echo, line feed, parity, and printer control. You are then responsible for the proper and efficient operation of the device controlled by this file. If you omit group and/or account names, the logon group and/or account name is substituted. For more information on terminal types, refer to appendix A, "Terminals Supported by MPE V/E", the <i>Fundamental Communications Handbook</i> (5957-4634) and the <i>Workstation Configurator Reference Manual</i> (30239-90001).
<i>cpusecs</i>	The maximum CPU time that your session can use, entered in seconds. When this limit is reached, the session is aborted. It may be any value from 1 to 32767, provided that it does not exceed the limit imposed by the system or account manager. To specify no limit, type a question mark (?) or UNLIM, or omit this parameter. The default is no limit.
BS,CS,DS, or ES	The priority queue that the command interpreter (CI) uses for executing your session. It is also the default priority queue for all programs executed within the session. BS is the highest priority, ES is the lowest, and the default is CS. If you specify a priority exceeding that permitted for your account or user name, MPE V/E assigns the highest priority possible below BS.

Caution



Use care in assigning the BS priority, because processes in this priority class may lock out other processes (MPE V/E uses the BS queue).

() COMMAND LOGON

DS and ES are intended primarily for batch jobs. Their use for sessions may produce less than optimum performance, and is generally discouraged.

For guidelines on the use of the priority queue, refer to the **TUNE** command in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

inputpriority or **HIPRI** Determines the input priority of the job or session. It may be a value from 1 (lowest priority) to 13 (highest priority). The default is 8. If you supply a value less than or equal to the current jobfence as set by the system operator, or if the jobfence is equal to or greater than the default **inputpriority**, the session is denied access.

You can use the **;HIPRI** option to override the system jobfence or to override the session limit. If you want to override the jobfence, the system first checks to see whether you have system manager (SM) or system supervisor (OP) capability. If you have either one, you are logged on, and your **;INPRI** defaults to the system's jobfence and execution limit. If you do not have either capability, you receive the following warning:

MUST HAVE "SM" OR "OP" CAP. TO SPECIFY HIPRI,
MAXIMUM INPRI OF 13 IS USED (CIWARN 1460)

Despite this warning, the system still allows you to logon if your default input priority, which is 8, exceeds the jobfence. If the jobfence exceeds your input priority, you cannot logon unless you have SM or OP capability.

Use

You may not use this command from a session, job, program, or **BREAK**. Pressing **BREAK** has no effect on this command.

If you press **BREAK** during the output of the logon message, the rest of the message is terminated and the command in parentheses is immediately executed. If you press **BREAK** while the command is being executed, and the command is breakable, you interrupt its execution. Any optional logon UDC that is not a program is not breakable, and its execution is forced during logon.

Operation

This command initiates a session and immediately executes the command in parentheses. Once this command or subsystem is executing, all input, output, **BREAK** capabilities, and prompts operate as described in the command or subsystem instructions. After the execution of the command or subsystem is completed, normally or through an abort, a logoff is performed automatically.

() COMMAND LOGON

This command has the following limitations:

- You must enter the () COMMAND LOGON at a terminal. You cannot use any other device to enter this command.
- Unlike the HELLO command, () COMMAND LOGON is not recognized within sessions.
- If you use a command that would normally generate an end-of-file (EOF), such as HELLO, BYE, JOB, DATA, :EOD, :EOF:, and EOJ, the session terminates as soon as the MPE V/E command interpreter encounters the command.
- Only one MPE V/E command is allowed within the parentheses, and leading and trailing blanks are suppressed.

If you omit any passwords required in the () COMMAND LOGON, MPE V/E prompts you for each one individually.

When MPE V/E initiates the session, it displays HP 3000, both the user-defined and official (base) *version.update.fix* number (an HP method of distinguishing between software releases), the date, and the time as follows:

```
HP3000  RELEASE:  G.03.08   USER VERSION: G.03.08   TUE, AUG 22, 1989,  1:28 PM
MPE V   HP32033  G.03.08   Copyright Hewlett-Packard 1984.  All Rights Reserved.
```

After the logon message is printed at your terminal, MPE V/E executes the command in parentheses. The information MPE V/E generates as a result of this command is transmitted to your terminal immediately, just as if you had called that command or subsystem during a session. When the command or subsystem is exited, MPE V/E terminates the session at once, and displays the CPU time used (in seconds), the connect time (in minutes), and the date and time, as follows:

```
CPU=1.  CONNECT=1.  TUE, JUL 11, 1989,  1:28 PM
```

Example

To logon under *username* MAC, *acctname* TECHPUB, and *groupname* XGROUP, and execute the LISTF command, enter:

```
:(LISTF) MAC.TECHPUB,XGROUP
```

```
HP3000  RELEASE:  G.03.08   USER VERSION: G.03.08   TUE, AUG 22, 1989,  1:28 PM
MPE V   HP32033  G.03.08   Copyright Hewlett-Packard 1984.  All Rights Reserved.
```

```
FILENAME
```

```
FIRST      GP005      TRIAL      XPROG
```

```
CPU=1.  CONNECT=1.  TUE, JUL 11, 1989,  1:28 PM
```

ABORT

Aborts the current program or operation.

Syntax

ABORT

Parameters

None.

Use

You may issue this command from a session, in **BREAK** only. You cannot issue it from a job or a program. Pressing **BREAK** has no effect on it.

Operation

After you suspend a program or MPE V/E command operation by pressing **BREAK**, the **ABORT** command terminates that program or operation. Programs do not terminate while critical system code is executing on their behalf, but will terminate immediately afterward. You can use the **ABORT** command only from a session and then only during a **BREAK**, but it does not disrupt the session. Some operations abort immediately on entering **BREAK** without requiring the **ABORT** command. When you issue an **ABORT** command, the job control word (JCW) is set to the "SYSTEM 0" state. For a discussion of job control words, refer to the **SETJCW** command in this chapter.

Example

To abort the current operation, enter:

:ABORT

ABORTIO/=ABORTIO

Aborts all pending I/O requests for a device, one at a time.

Syntax

ABORTIO *ldev*

=ABORTIO *ldev*



Parameters

ldev The logical device number of the device that has the pending I/O request(s).

Use

You may issue this command from a session or job, from within **BREAK** or from a program. Pressing **BREAK** has no effect on it. You can execute this command from the console only *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** command.

Operation

This command aborts all pending I/O operations on the specified device. To delete all queued I/O requests for a device, repeat the **ABORTIO** command until the following message is printed on the **\$STDLIST** device:

NO I/O TO ABORT FOR DEVICE #ldev

Those devices that accept **JOB** or **DATA** always have **READ** requests pending, due to the auto-recognition feature of MPE V/E. The **ABORTIO** command is used to clear these pending input requests.

To completely clear spooled devices, you may need several **ABORTIO** commands. Issue the **ABORTIO** command until there are no queued I/O requests, and the message above appears on the **\$STDLIST** device.

Sometimes you need to clear all outstanding I/O requests so that other console commands can be properly executed. In certain cases, the **ABORTJOB**, **TAKE**, **DOWN**, and **REFUSE** commands will not execute unless you first issue an **ABORTIO** command to clear pending I/O operations on the appropriate device.

If **ABORTIO** is not effective from the system console, **=ABORTIO** may be used.

Note



The **=ABORTIO** command should be used only when the **ABORTIO** command cannot be executed, or when the console is busy. **CTRL** A followed by the **=ABORTIO** command can be issued only from the physical console unless distributed to users with the **ALLOW** or **ASSOCIATE** command.

ABORTIO/=ABORTIO

Examples

To abort all pending I/O requests for logical device #53, enter:

```
:ABORTIO 53  
11:16/3/NO I/O TO ABORT FOR DEVICE 53
```

You may have to issue several ABORTIO commands to abort all pending I/O operations on a spooled device, as shown below:

```
:STOPSPool 5  
11:20/31/SP#5/STOPPED  
11:20/31/LDEV#5 NOT READY  
:REFUSE 5  
:ABORTIO 5  
:ABORTIO 5 IS THIS CORRECT?  
11.21/40/NO I/O TO ABORT FOR DEVICE 5
```

ABORTJOB/=ABORTJOB

Aborts a job or session.

Syntax

$$\begin{array}{l} \text{ABORTJOB} \left\{ \begin{array}{l} \#Jnnn \\ \#Snnn \\ [jobname] user.acct \end{array} \right\} \\ \\ \text{=ABORTJOB} \left\{ \begin{array}{l} \#Jnnn \\ \#Snnn \\ [jobname,] user.acct \end{array} \right\} \end{array}$$

Parameters

#Jnnn A job number.
#Snnn A session number.
jobname The name of the job, as identified by the SHOWJOB command.
user A user name.
acct An account name.

Use

You can issue this command from a session, job, program or in **BREAK**. Pressing **BREAK** has no effect on this command. You can execute it only:

- from the console, or
- if it is distributed to users with the **ALLOW** command, or
- if **JOBSECURITY** is set to **LOW** and it is your own job, or
- if **JOBSECURITY** is set to **LOW** and you have account manager (AM) or system manager (SM) capabilities.

You may issue the **=ABORTJOB** command only from the console.

Operation

This command terminates the designated job or session and displays the following message on the job/session list device:

SESSION ABORTED BY SYSTEM MANAGEMENT

If you use the "[*jobname*,]*user.acct*" form of the command when there is more than one job or session executing under that name, MPE V/E randomly selects which job or session to abort. Therefore, to exercise more precise control when aborting jobs or sessions, use the **#Jnnn** or **#Snnn** form of the **ABORTJOB** command. Although the job or session is abnormally terminated, the system issues log records and updates CPU and connect times. Any I/O activity (for example, printing or **STORE**) is also terminated.

ABORTJOB/=ABORTJOB

You can apply the **ABORTJOB** command to waiting and scheduled jobs, as well as to executing jobs. If the spooler input file (**\$STDIN**) for a batch job has been created but not yet opened (that is, the job is in the **WAIT** state), the entire file is deleted. If you issue the **ABORTJOB** command before the output spool file is complete, only the portion of the file already spooled is printed, along with an error message indicating that the job was aborted.

If a request is pending at the system console, it is terminated automatically by the **ABORTJOB/=ABORTJOB** command. The following message appears on the system console:

```
time/#Snnn/pin/REQUEST REQUIRING OPERATOR REPLY FOR PIN #nn HAS BEEN ABORTED
```

When the **ABORTJOB** command is successful, a logoff message is displayed, indicating that the job being executed has been aborted. This is shown in the example below:

```
:ABORTJOB #S9
11:20/#S9/34/LOGOFF ON LDEV #77
```

The system also displays two messages on the user's terminal: the standard error message that appears when a request is manually terminated by a **=REPLY** of 0 or N, and the following:

```
SESSION ABORTED BY SYSTEM MANAGEMENT
```

The **=ABORTJOB** command may be used at the console if **ABORTJOB** is ineffective. Refer to the "USE" section of this command.

Examples

To terminate session #139, enter:

```
:ABORTJOB #S139
17:10/#S139/34/LOGOFF ON LDEV #62
```

To terminate job #9, enter:

```
:ABORTJOB #J9
20:18/#J9/26/LOGOFF ON LDEV #10
```

Note that in both cases, the execution was successful, and a logoff message was printed.

To terminate session #6, which has a pending device allocation message, enter:

```
?17:00/#S6/23/LDEV# FOR "SCRTAPE" ON TAPE (NUM)?
:ABORTJOB #S6
17:10/#S6/120/REQUEST REQUIRING OPERATOR REPLY FOR PIN 23
HAS BEEN ABORTED
17:10/#S6/120/LOGOFF ON LDEV #58
```

ACCEPT

Permits a designated device to accept jobs, sessions and/or data.

Syntax

```
ACCEPT [ JOBS,
        DATA, ] ldev
```

Parameters

JOBS	The designated device recognizes the JOB and HELLO commands. The device must be interactive to support sessions.
DATA	The designated device recognizes the DATA command.
<i>ldev</i>	The logical device number of the device for which the JOB, HELLO , and/or DATA commands are being enabled.

Note



If you omit both the JOBS and the DATA parameters, then the JOB, HELLO and DATA commands are all allowed on the designated device.

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** has no effect on this command. You can execute it only from the console *unless* it is distributed to users with the ALLOW or ASSOCIATE commands.

Operation

The system operator or system supervisor uses this command to designate which devices may be used to initiate jobs, sessions and/or data. When a device is configured as an “accepting” device, MPE V/E automatically scans the first input record for a valid JOB, HELLO, or DATA command. This feature, called “auto-recognition”, allows you to access the device without specifically requesting use of the device via a message to the system console.

If the JOBS parameter is explicitly requested, the ACCEPT command is not executed unless the device has been configured with a default output device. For more information, refer to “Reconfiguring the System” in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

ACCEPT

Examples

To permit logical device #19 to accept jobs and data, enter:

```
:ACCEPT 19
:SHOWDEV 19
LDEV  AVAIL      OWNERSHIP      VALID      ASSOCIATION
```

```
19 A AVAIL
```

To permit logical device #19 to accept jobs and data, and to allow the device to be spooled, enter:

```
:ACCEPT 19
:STARTSPOOL 19
11:12/31/SP#/SPOOLED IN
11:12/6/LDEV#19 NOT READY
:SHOWDEV 19
DEV  AVAIL      OWNERSHIP      VALID      DEN      ASSOCIATION

19  SPOOLED      SPOOLER OUT
```

ALLOCATE

Permanently loads a program or procedure into main memory.

Syntax

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

Parameters

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

Use

You may issue this command from a session or job. You cannot issue it from a program, or in **BREAK**. Pressing **(BREAK)** has no effect on it. You must have system supervisor (OP) capability to use this command.

Operation

A program or procedure is allocated by resolving external references and assigning code segment table (CST) or extended code segment table (XCST) entries to the program's code segments. Table entries are also allocated for any procedures called by the allocated program or procedure. Allocating a program or procedure does not increase execution speed. However, it does reduce the time taken to load the program for execution.

Caution



Use care in deciding which programs or procedures to load with **ALLOCATE**. The number of CST table entries is limited. If you exceed the limit, you run the risk of losing data.

Segments remain loaded until they are deallocated with the **DEALLOCATE** command, or until the system is shut down or a system failure occurs. You must reallocate programs or procedures with the **ALLOCATE** command following any start-up.

You must have **EXECUTE** access for any file referenced in the *name* parameter of the **ALLOCATE** command if you want to issue this command.

If a program is run with the **LIB={G,P, or S}** parameter and you allocate it while it is still running, the **LIB** parameter is allocated as well. When you run this program subsequently, the allocated **LIB** parameter is used.

ALLOCATE

Example

To allocate a procedure identified as PROC1, that resides in SL.PUB.SYS, enter:

```
:ALLOCATE PROCEDURE,PROC1
```

Note



Program files residing in the non-system domain (a private volume) are not allocated. If you attempt to allocate them, you will receive “LOAD ERR 92 ”, which reads as follows: **Attempting to allocate or deallocate program from non-system disk.**

Additional Discussion

Refer to the discussion of program allocation in “Changing Other System Parameters” in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

ALLOW

Grants a user access to a specific operator command.

Syntax

ALLOW FILE=*formaldesignator* [;SHOW]

$$\text{ALLOW} \left\{ \begin{array}{l} @.@ \\ \text{user}.@ \\ @.acct \\ \text{user}.acct \end{array} \right\}; \text{COMMANDS}=\text{command} [, \text{command}, \dots]$$

Parameters

<i>formaldesignator</i>	A formal ASCII filename, which may consist of one to eight alphanumeric characters, beginning with an alphabetic character.
SHOW	Lists input lines on \$STDLIST.
@.@	Grants access to all users whether logged on or not.
user.@	Grants access to a specific user in all accounts.
@.acct	Grants access to all users in a specific account.
user.acct	Grants access to a specific user in a specific account.
command	Names the commands to which the user is granted access.

Use

You can issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You can execute it only from the console *unless* it is distributed to users with the **ALLOW** command.

Operation

The system operator uses the **ALLOW** command to distribute specific operator commands to system users. You indicate at the console which users may execute operator command(s), and which commands they are allowed to execute.

This command may also be executed in indirect and subsystem modes. You must create a file that contains records identifying the user(s) and account(s) to whom you are allowing the use of system operator commands, followed by the list of commands allowed.

ALLOW

For example:

```
:EDITOR
HP32201A.7.16 EDIT/3000  TUES, JUL 11, 1989,  5:08 PM
(C) HEWLETT-PACKARD CO. 1984
/ADD
  1  SUSAN.PAYROLL;COMMANDS=ALTJOB,ALTSPoolFILE
  2  JOHN.ACCTNG;COMMANDS=ALTSPoolFILE,DELETESPoolFILE
  3  //
...
/KEEP ALLOWTMP
/E
```

Then issue the **ALLOW** command, using the **;SHOW** parameter to display each command line as it is executed from the file.

```
:ALLOW FILE=ALLOWTMP;SHOW
```

In subsystem mode, enter “**ALLOW**”, followed by Return. You are prompted with a “>”. Command parameters are accepted until an end-of-file (EOF) is received, or until you exit.

The system operator may allow console capabilities only to users who are currently logged on to the system, unless the **@.@** option of the **ALLOW** command is used. In this case, all users are affected, whether or not they are logged on. Additional capabilities granted to a user last only for the duration of the current session. Once the user logs off, any special capabilities previously assigned no longer apply.

The user command **SHOWALLOW** indicates the operator commands that have been allowed globally and to a specific user. Refer to **SHOWALLOW** in this chapter.

Examples

To give the user **MAC.TECH** the ability to execute the **REPLY** and **ABORTIO** commands, enter the following on the system console:

```
:ALLOW MAC.TECH;COMMANDS=REPLY,ABORTIO
```

In subsystem mode, to give the user **MGR.MANUALS** the ability to execute the **BREAKJOB** command, enter the following on the system console:

```
:ALLOW
>MGR.MANUALS;COMMANDS=BREAKJOB
>EXIT
```

ALTACCT

Changes the attributes of an existing account.



Syntax

```
ALTACCT acctname [ ;PASS=[password] ] [ ;FILES=[filespace] ] [ ;CPU=[cpu] ]
[ ;CONNECT=[connect] ] [ ;CAP=[capabilitylist] ] [ ;ACCESS=[(fileaccess)] ]
[ ;MAXPRI=[subqueueuname] ] [ ;LOCATTR=[localattribute] ] [ ;VS=[volset: { ALT
SPAN } ] ]
[ ;USERPASS=[ { REQ
OPT } ] ] (1)
```

(1) The USERPASS parameter is only available if the HP Security Monitor (HP30392) has been installed.

Parameters

<i>acctname</i>	The name of the account to be altered.
<i>password</i>	Account password (used only for verifying logon access). If you omit this parameter, no change is made. If the system operator enters “;PASS= Return ”, the existing password is removed.
<i>filespace</i>	Disk storage limit, in sectors, for the permanent files in the account. The <i>filespace</i> limit cannot be less than the number of sectors currently in use for the account. The default is unlimited file space.
<i>cpu</i>	The limit on cumulative CPU time, in seconds, for the account. This limit is checked only when a job or session is initiated, and therefore never causes the job or session to abort. The maximum value allowed is 2,147,483,647 seconds. The default is unlimited CPU time. The counter may be set to zero with the RESETACCT command.
<i>connect</i>	The limit on total cumulative job or session connect time, in minutes, for the account. This limit is checked at logon and every time a process terminates. The maximum value allowed is 2,147,483,647 minutes. The default is unlimited connect time. The counter may be set to zero with the RESETACCT command.

ALTACCT

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

System Manager	= SM
Account Manager	= AM
Account Librarian	= AL
Group Librarian	= GL
Diagnostician	= DI
System Supervisor	= OP
Network Administrator	= NA
Node Manager	= NM
Permanent Files	= SF
Access to nonsharable I/O devices	= ND
Use Volumes	= UV
Create Volumes	= CV
Use Communications Subsystem	= CS
Programmatic Sessions	= PS
User Logging	= LG
Process Handling	= PH
Extra Data Segments	= DS
Multiple RINs	= MR
Privileged Mode	= PM
Interactive Access	= IA
Local Batch Access	= BA
Process Handling	= PH

The default is “AM,AL,GL,SF,ND,IA,BA” , except for the SYS account. This account has no true default: it is assigned the maximum account capabilities when the system is delivered and, under normal circumstances, should not be altered. CV capability, which permits account members to create private volumes, automatically gives the account UV capability, allowing account members to use private volumes as well.

<i>fileaccess</i>	The restrictions on file access pertinent to this account. The default is R,A,L,W,X:AC, entered as follows: $\left[\left[\left(\begin{array}{c} R \\ L \\ A \\ W \\ X \end{array} \right) [, \dots] : \left\{ \begin{array}{c} ANY \\ AC \end{array} \right\} [; \dots] \right] \right]$ where R , L , A , W and X specify modes of access by types of users (ANY and/or AC) as follows: R = READ L = LOCK (allows exclusive access to file) A = APPEND (implicitly specifies L also) W = WRITE (implicitly specifies A and L also) X = EXECUTE The user types are specified as follows: ANY = Any user AC = Member of this account only
<i>subqueue</i> <i>name</i>	Name of the highest priority subqueue that can be requested by any process of any job or session in the account, specified as AS, BS, CS, DS or ES. The default is CS .

Caution

User processes executing in the AS or BS subqueues can deadlock the system. If you assign these subqueues to non-priority processes, other critical system processes may be prevented from executing. Exercise extreme caution when choosing subqueues.

<i>localattribute</i>	Local attribute of the account, as defined at the installation site. This is a double-word bitmap, of arbitrary meaning, that might be used to further classify accounts. Although it is not part of standard MPE V/E security provisions, it is available to processes through the WHO intrinsic. Programmers may use <i>localattribute</i> in their own programs to provide security. The default is double-word 0 (null).
<i>volset</i>	The volume set or class reference which, when fully qualified, is in the form <i>vcsid.groupname.acctname</i> . <i>vcsid</i> refers to a previously defined volume set or class definition.

ALTACCT

SPAN	Indicates that the <i>acctname</i> is to be inserted in the accounting directory of the specified volume set (<i>volset</i>). The specified volume set must already be mounted (via a MOUNT command) for the SPAN operation to succeed.
ALT	Directs the altering of an account or group entry on the specified volume set. This option is useful only if it is necessary to alter account and group file space limits for entries that have already been spanned.
REQ	USERPASS=REQ specifies that all users in the account must have a non-blank password. It is available only if the HP Security Monitor (HP30392) has been installed. Refer to the <i>HP Security Monitor User's Guide</i> for more information.
OPT	USERPASS=OPT specifies that users in this account may or may not have passwords. If you do not use the USERPASS parameter, the old value remains. It is available only if the HP Security Monitor has been installed.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have system manager (SM) capability to use this command.

Operation

The system manager uses **ALTACCT** to change the attributes of an existing account. You may enter multiple keywords on a single command line, as shown in the examples. When you change one capability in a *capabilitylist* containing several defined values, you must respecify the entire *capabilitylist*. 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.

The **USERPASS=** parameter must be used in conjunction with **REQ** or **OPT**. The old value remains in effect if the **USERPASS** parameter is omitted. This is only valid if you have installed the HP Security Monitor.

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

Any value changed with ALTACCT takes effect the next time MPE V/E is requested to check the value. If an attribute is removed from an account while users are logged on, they are not affected until they end the job or session and logon again.

MPE V/E does not automatically generate a message informing users of the change; it is the system operator's responsibility to warn account members in advance of any changes. If the system operator takes a capability away from an account, all account members and groups within the account also lose the capability.

You cannot remove system manager (SM) capability from the SYS account. You also cannot take AM capability away from any account. If more than one person is assigned AM capability within the account, you can remove AM capability from all but one of the users assigned it, that is, the last.

Example

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

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

For additional information, refer to the *HP Security Users Guide* (30392-90001)

ALTGROUP

Changes one or more attributes of a group.

Syntax

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

Parameters

- groupname* Specifies the name of the group whose attributes are to be changed.
- accountname* As system manager you can include the account name when you are logged onto an account other than the requested account.
- password* The new password to be assigned to the group (used to verify logon access only). If this parameter is omitted, no change is made. If the system operator enters “;PASS=Return”, then the existing password is removed. The password must consist of one to eight alphanumeric characters, beginning with an alphabetic character.
- capabilitylist* Specifies the list of capabilities permitted this group. Each capability is denoted by a two-letter mnemonic, as follows.
- | | | |
|---------------------|---|----|
| Process Handling | = | PH |
| Extra Data Segments | = | DS |
| Multiple RINs | = | MR |
| Privileged Mode | = | PM |
| Interactive Access | = | IA |
| Local Batch Access | = | BA |
- Two or more capabilities may be specified if they are separated by commas.
- The default is IA,BA except for the PUB group of the SYS account. This group has no true default. It is assigned the maximum group capabilities when the system is delivered and should not normally be changed.
- filespace* Disk storage limit, in sectors, for the permanent files of the group. A group's *filespace* cannot be set to a value greater than the corresponding limits currently defined for the group's account. If an account's *filespace* is later changed, the group limits are automatically changed. The default is unlimited file space.

cpu The limit on the total cumulative CPU time, in seconds, for the group. This limit is checked only when a job or session is initiated; the limit never causes a job or session to abort. The maximum value you may specify with this command is 2,147,483,647 seconds. If the limit is exceeded, users with account manager capability are warned when logging on; other users are denied access.

The CPU limit for a group cannot be set to a value greater than the limit currently defined for the account in which that group resides. If the CPU limit for an account is changed later, group limits are automatically changed. The default is unlimited CPU time. The counter may be set to zero with the **RESETACCT** command.

connect The limit on the total cumulative job or session connect time, in minutes, that the group is allowed. This limit is checked at logon, and whenever the job or session initiates a new process. The maximum value you may specify with this command is 2,147,483,647 minutes. If the limit is exceeded, users with AM (account manager) capability are warned when logging on; other users are denied access.

A group's connect limit cannot be specified as greater than the limit currently defined for the account in which the group resides. However, if the connect limit of the account is changed later, the limits of its component groups are automatically changed. The default is unlimited connect time. The counter may be set to zero with the **RESETACCT** command.

fileaccess The restriction on file access pertinent to this group. The default is **R,X:ANY;A,W,L,S:AL,GU** for the public group (PUB); and **R,A,W,L,X,S:GU** for all other groups, entered as follows:

$$\left\{ \begin{array}{c} R \\ L \\ A \\ W \\ X \\ S \end{array} \right\} [, \dots] : \left\{ \begin{array}{c} ANY \\ AC \\ GU \\ AL \\ GL \end{array} \right\} [, \dots]$$

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

R	=	READ
L	=	LOCK (allows exclusive access to the file)
A	=	APPEND (implicitly specifies L also)
W	=	WRITE (implicitly specifies A and L also)
X	=	EXECUTE
S	=	SAVE

ALTGROUP

The user types are specified as follows:

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

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

volset A volume set or class reference which, when fully qualified, is in the following format:

vcsid.groupname.acctname

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

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

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

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

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

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

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** has no effect on this command. You must have account manager (AM) capability to use this command.

Operation

This command changes one or more attributes of a group; you may specify multiple parameters on a single command line as shown in the example. When you omit an entire parameter from an ALTGROUP command, the corresponding value for the group remains unchanged. When you include a keyword but omit the corresponding parameter (as in ;PASS=), the default value is assigned as follows:

Parameter	Default Values
<i>password</i>	Null (No password)
<i>capabilitylist</i>	IA,BA (except PUB.SYS) PH,DS,MR,PH,IA,BA (PUB.SYS only)
<i>filespace</i>	Unlimited
<i>cpu</i>	Unlimited
<i>connect</i>	Unlimited
<i>fileaccess</i>	R,A,W,L,X,S:GU (All groups except PUB) R,X:ANY;A,W,L,S:AL,GU (PUB group only)

When a parameter is modified through the ALTGROUP command, it immediately takes effect in the directory. It does not affect any active users with open files in the group until they terminate their session and logon to that user name and group again. For this reason, it is a good idea to notify all group users of any planned changes in advance.

Example

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

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

ALTJOB

Alters the attributes of waiting or scheduled jobs.

Syntax

$$\text{ALTJOB } \left\{ \begin{array}{l} \#Jnnn \\ \#Snnn \end{array} \right\} \left[; \text{INPRI} = \text{inputpriority} \right] \left[; \text{OUTDEV} = \left\{ \begin{array}{l} ldev \\ devclass \end{array} \right\} \right]$$

Parameters

#Jnnn	A job number.
#Snnn	A session number. (Although syntactically correct, this parameter is rarely used; sessions do not wait.)
inputpriority	Specifies the new input priority. 0 is the lowest; 14 is the highest.
ldev or devclass	Specifies the logical device number or device class name of the job's \$STDLIST on the destination device.

Use

You may issue this command from a session, job, program or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute **ALTJOB** to change your own jobs. To change other users' jobs, the command must be issued from the console *unless* it is distributed to users with the **ALLOW** command, or **JOBSECURITY** is set to **LOW**.

Operation

The **ALTJOB** command, in conjunction with the **JOBFENCE** command, allows you to control the flow of all jobs on the system with the exception of **HIPRI** jobs. You can use it to alter jobs in the **INTRO**, **WAIT** and **SCHED** states only. Jobs with an input priority lower than or equal to the current jobfence, a numerical value from 0 to 14, are deferred.

Example

Consider three jobs that are submitted by a user and suppose that all of these jobs have an **INPRI** value of 8 . To change their **INPRI** values to ensure that **JOB1** runs first, **JOB2** runs last, and **JOB3** runs second with **LP** allocated as the **OUTDEV** for **JOB3**, enter the following commands:

```
:JOBFENCE 14
15:11/#J1/24/DEFERRED JOB INTRODUCED ON LDEV #10  **Prevents all jobs**
15:11/#J2/25/DEFERRED JOB INTRODUCED ON LDEV #10  **from running while **
15:13/#J3/26/DEFERRED JOB INTRODUCED ON LDEV #10  **the user changes**
                                         **the INPRI values.**
```

:SHOWJOB

JOBNUM	STATE	IPRI	JIN	JLIST	INTRODUCED	JOB NAME
#S23	EXEC		20	20	THU 2:15P	OPERATOR.SYS
#J1	WAIT	D 8	10S	12	THU 3:11P	JOB2,OPERATOR.SYS
#J2	WAIT	D 8	10S	12	THU 3:11P	JOB3,SARAH.PAYROLL
#J3	WAIT	D 8	10S	12	THU 3:13P	JOB1,JOHN.ACCTNG

4 JOBS:

0 INTRO

3 WAIT; INCL 3 DEFERRED

1 EXEC; INCL 1 SESSIONS

0 SUSP

JOBFENCE= 14; JLIMIT= 5; SLIMIT= 16

:ALTJOB #J1;INPRI=10

:ALTJOB #J3;INPRI=9;OUTDEV=LP

:ALTJOB #J2;INPRI=8

:JOBFENCE 8

16:21/#J1/34LOGON FOR: JOB1,FIELD.SUPPORT ON LDEV #10

16:21/#J3/35LOGON FOR: JOB3,FIELD.SUPPORT ON LDEV #10

:SHOWJOB

JOBNUM	STATE	IPRI	JIN	JLIST	INTRODUCED	JOB NAME
#S23	EXEC		20	20	THU 2:15P	OPERATOR.SYS
#J1	EXEC	D10	10S	12	THU 3:11P	JOB1,OPERATOR.SYS
#J2	EXEC	D 9	10S	LP	THU 3:13P	JOB3,JOHN.ACCTNG
#J3	WAIT	D 8	10S	12	THU 3:11P	JOB2,SARAH.PAYROLL

4 JOBS:

0 INTRO

3 WAIT; 3 DEFERRED

1 EXEC; INCL 1 SESSIONS

0 SUSP

JOBFENCE=8; JLIMIT=5; SLIMIT=16

If you reset the jobfence at 6, all jobs with INPRI values of 7 or above are run.

ALTLOG

Alters the attributes of an existing identifier for logging users.

Syntax

$$\text{ALTLOG } \textit{logid} \left[; \text{LOG}=\textit{logfile} \left\{ \begin{array}{l} , \text{DISC} \\ , \text{TAPE} \\ , \text{SDISC} \\ , \text{CTAPE} \end{array} \right\} \right] \left[; \text{PASS}=\textit{password} \right] \left[\left\{ \begin{array}{l} ; \text{AUTO} \\ ; \text{NOAUTO} \end{array} \right\} \right]$$

Parameters

<i>logid</i>	The logging identifier whose attributes are to be changed. This identifier must contain from one to eight alphanumeric characters, beginning with an alphabetic character.
<i>logfile</i>	The name of the file to receive data from the logging procedure. This name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. You must specify the device class on which <i>logfile</i> resides, that is, DISC, TAPE, SDISC, or CTAPE.
<i>password</i>	The new password for the logging identifier. This password must contain from one to eight alphanumeric characters, beginning with an alphabetic character.
AUTO	Initiates an automatic CHANGELOG if the current log file becomes full. Refer to CHANGELOG in this chapter.
NOAUTO	Prevents the initiation of an automatic CHANGELOG . A CHANGELOG is not automatically performed when the current log file becomes full. This is the default.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. It requires system supervisor (OP) or user logging (LG) capability.

Operation

This command changes the attributes of an existing identifier for logging users to those specified in the parameter list. Parameters not included in the **ALTLOG** command retain their current values. You must have user logging (LG) capability to execute **ALTLOG**. Only the creator of the logging identifier can alter its attributes.

To use the **AUTO** parameter, the log process for *logid* must be enabled for changing. You may do this by ending the log filename with the numeric characters "001" (for example, *.fname001*). This naming convention works in conjunction with the fileset number to generate sequential filenames automatically.

Note

If a new log filename is specified with the ALTLOG command, the links with any previous log file are broken.

Example

To change the destination *logfile* of the logging identifier **KIM** to *logfile C* and specify that **C** resides on disk, enter:

```
:ALTLOG KIM;LOG=C,DISC
```

Since the keyword parameter **PASS=** was not included, **KIM** retains any password previously specified.

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005)

MPE V Intrinsic Reference Manual (32033-90007)

ALTSEC

Alters a file's security provisions and manipulates the access control definitions (ACDs) associated with files and devices.

Syntax

$$\text{ALTSEC } \textit{objectname} \left[, \left\{ \begin{array}{l} \text{FILENAME} \\ \text{LDEV} \\ \text{DEVCLASS} \end{array} \right\} \right] ; \{ [\text{ACCESS=}] (\textit{fileaccess} [; \textit{fileaccess} [; \dots]]) \}$$
$$\left\{ \begin{array}{l} \text{NEWACD} = \left\{ (\textit{acd pair} [; \textit{acd pair} [; \dots]]) \right\} \\ \quad \quad \quad \textit{filereference} \\ \text{COPYACD} = \left\{ \textit{objectname} \left[, \left\{ \begin{array}{l} \text{FILENAME} \\ \text{LDEV} \end{array} \right\} \right] \right\} \\ \text{ADDPAIR} = \left\{ (\textit{acd pair} [; \textit{acd pair} [; \dots]]) \right\} \\ \quad \quad \quad \textit{filereference} \\ \text{REPPAIR} = \left\{ (\textit{acd pair} [; \textit{acd pair} [; \dots]]) \right\} \\ \quad \quad \quad \textit{filereference} \\ \text{DELPAIR} = \left\{ (\textit{user spec} [; \textit{user spec} [; \dots]]) \right\} \\ \quad \quad \quad \textit{filereference} \end{array} \right\} \{ \text{DELACD} \}$$

Parameters

objectname Specifies an actual file designator (FILENAME), a logical device number (LDEV), or a device class name (DEVCLASS). The default is FILENAME.

For files, the actual designator of the file for which security provisions are to be altered, is written in the format:

filename[/lockword][.groupname[.acctname]]

If a lockword exists, you must specify it when you create a batch job. If a lockword exists and you omit it from a session, MPE V/E prompts you for it. If you do not specify *groupname.acctname*, MPE V/E assumes the logon group and account.

If a logical device number (LDEV) is specified, it must be configured in the system device configuration.

If a device class name (DEVCLASS) is specified, it must be configured in the system device configuration.

ACCESS Identifies the alteration to the file access specifications at the file level.

fileaccess

File security specifications, entered as **R**, **L**, **A**, **W**, and **X** which specify modes of access, and **ANY**, **AC**, **GU**, **AL**, and **GL** which specify types of users, are as follows:

$$\left\{ \begin{array}{c} \mathbf{R} \\ \mathbf{L} \\ \mathbf{A} \\ \mathbf{W} \\ \mathbf{X} \end{array} \right\} [, \dots] : \left\{ \begin{array}{c} \mathbf{ANY} \\ \mathbf{AC} \\ \mathbf{GU} \\ \mathbf{AL} \\ \mathbf{GL} \end{array} \right\} [, \dots]$$

All modes are accepted for both files and devices. If you specify a mode that is not applicable to an object, the system accepts it but does not use it for access checking. Two or more modes may be specified if they are separated by commas. The mode types are specified as follows:

R = READ
L = LOCK (allows opening with dynamic locking option)
A = APPEND (implicitly specifies **L** also)
W = WRITE (implicitly specifies **A** and **L** also)
X = EXECUTE

Note

ALTSEC works in the same way for devices as well as files. You may specify all the modes; however, depending on the type of device, some of the modes may not be used in access checking.

Two or more user types may be specified if they are separated by commas. The default is **R**, **L**, **W**, **A**, **X**: **ANY**. The colon (:) separating one or more modes from one or more user types is required when you specify *fileaccess*.

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

For example:

ACCESS = (R,W:ANY;A,L,X:CR)

NEWACD

Identifies the addition of a new access control definition (ACD) to an object.

acd pair

An *acd pair* is specified in the same manner as *fileaccess* (see above). Both *modes* and *user spec* specification must be supplied.

ALTSEC

The *acd pair* is specified in the form:

modes : user spec

If *acd pair* is specified in a command line, separate the pairs with a semicolon (;) and enclose the entire ACD specification within parentheses. You may have no more than 20 ACD pairs in a file.

modes Specifies types of access and permissions. This can be a combination of any of the following types of access permissions, separated by a comma:

```
R      : READ
W      : WRITE
L      : LOCK
A      : APPEND
X      : EXECUTE
NONE   : No Access
RACD   : Copy or read the ACD permission
```

user spec Specifies a fully qualified user name defined by *username.accountname*. You separate multiple user names by a comma. You can substitute wildcards for the user name or for both the user name and the account name, as follows:

@.accountname : Represents all users in *accountname*

@.@ : Represents all users in the system.

If no match is found in the ACD for a user, that user is prevented from accessing the object.

^filereference The name of the file containing *acd pairs*. The *acd pairs* may be placed on separate lines, but a single pair may not span more than one line. The *acd pair* may be enclosed within parentheses. For example,

```
(R,W: JOE.BLOW; R: @.BLOW;      ** Record 1 - first ACD pair **
 X: @.@)                        ** Record 2 - second ACD pair **
```

The ^ indicates that the contents of the file are used, not the name.

The contents of the indirect file must contain syntactically correct *acd pairs* enclosed within parentheses. A file reference has the following format:

filename[/lockword][.groupname[.acctname]]

If the file has a lockword, it must be specified. If *groupname.accountname* is not specified, MPE V/E assumes the logon group and account. System files such as \$STDIN are not allowed.

COPYACD Identifies the filename or *ldev* whose ACD is to be copied.

ADDPAIR Identifies the ACD pairs to be added to an existing ACD.

REPPAIR Identifies an ACD pair which will replace an existing ACD.

DELPAIR Identifies the ACD pair to be deleted. Only a *user spec* need be specified in DELPAIR.

DELACD Deletes all access control definitions from an object.

2-38 Command Definitions

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** aborts execution.

Operation

The **ALTSEC** command lets you manipulate access control definitions (ACDs) associated with files and devices.

ALTSEC enables you to establish or alter the security provisions for any disk file. It permanently deletes all provisions specified for a file at the file level, and replaces them with those defined by the current **ALTSEC** parameters. This operation is done by defining the new file-level access provisions in parentheses. It is recommended that the **ACCESS** keyword be used to identify the file access operation.

To manipulate ACDs with **ALTSEC**, you use one of the descriptive keywords associated with ACD operations, such as **NEWACD**, **COPYACD**, **ADDPAIR**, **REPPAIR**, **DELPAIR** or **DELACD**.

If an ACD exists for a file, it takes precedence over any file access restrictions added with the **ACCESS** keyword.

Example

You have created a file named **FDATA** and you want to change its security provisions to allow **WRITE** access to yourself only. There are no default security provisions. Enter:

```
:ALTSEC FDATA;ACCESS=(W:CR)
```

To change the security provisions for a program file (**FPROG**) so that any group user can execute the program, but only account and group librarian users can read or write to the file, enter:

```
:ALTSEC FPROG;ACCESS=(X:GU;R,W:AL,GL)
```

To set access modes and associate an ACD with a file or device, enter:

```
: ALTSEC APPA; NEWACD=~ACDINFO
```

The file **ACDINFO** contains the ACD information which is now associated with the file **APPA**. The content of **ACDINFO** file is:

```
R :MANAGER.SYS;
X :USER1 .ACCOUNT1
```

Additional Discussion

MPE V Intrinsic Reference Manual (32033-90007)

MPE File System Reference Manual (30000-90236)

MPE V System Operation and Resource Management Reference Manual (32033-90005)

MPE V Account Structure and Security (32033-90136)

ALTSPoolFILE

Alters the characteristics of an output spool file.

Syntax

$$\text{ALTSPoolFILE } \left\{ \begin{array}{l} \#0nnn \\ ldev1 \end{array} \right\} \left\{ \begin{array}{l} ;\text{PRI}=\text{outputpriority} \\ ;\text{COPIES}=\text{numcopies} \\ ;\text{DEV}=\left\{ \begin{array}{l} ldev2 \\ devclass \end{array} \right\} \\ ;\text{DEFER} \end{array} \right\} [; \dots]$$

Parameters

#0nnn	The output device file identification of a spool file.
<i>ldev1</i>	The logical device number of the device where an active spool file currently resides.
<i>outputpriority</i>	Specifies the output priority of the designated device file. 0 is the lowest; 14 is the highest.
<i>numcopies</i>	Specifies the number of copies to be produced from the designated device file. The range is 1 through 127; the default is 1.
<i>ldev2</i> or <i>devclass</i>	Specifies the logical device number or device class name of the spool file's destination device. If active, the file is returned to the ready state. It may immediately become active on <i>ldev2</i> if all requirements are met.
DEFER	Immediately changes the output priority of an active or ready spoolfile to 0 . If active, the file is returned to the ready state.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You can execute it from the console if you have OP or SM capability: it may also be distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The system operator uses **ALTSPoolFILE** to change the printing priority of a spool file, to increase or decrease the number of copies produced, and to change the destination device or class.

When altering an active spool file, first take the output device offline. This gives you time to enter the command and determine that the active spool file is the file being printed. When you have sent the **ALTSPoolFILE** command to the spooler, MPE V/E returns the colon prompt (:). No change to the spool file is made, however, until the output device is shown to be online.

You may alter the *outputpriority* or the *numcopies* of an active spool file without interrupting the printing process. If you alter the device or defer the active spool file with the **DEFER** parameter, the printer stops immediately. In both of these cases, the entire file is printed when printing resumes. Deferring a spool file lowers its output priority to zero, the lowest priority possible. To print a deferred spool file, you must raise its priority above the current outfence using the **ALTSPoolFILE** command.

If you intend to print a spool file on an HP 2680A Laser Printer, you may add an environment file to it before printing. Refer to the discussion of the **SPool >COPY** command in the *MPE V System Utilities Reference Manual* (32033-90008) for more information.

Examples

To defer the active spool file (#086) on *ldev* 6, take logical device #6 offline, then enter:

```
:ALTSPoolFILE #086;DEFER
```

or

```
:ALTSPoolFILE 6;DEFER
```

To change the priority of deferred spool file #0123 from 0 to 3, enter:

```
:ALTSPoolFILE #0123;PRI=3
```

ALTUSER

Alters attributes currently defined for a user.

Syntax

```
ALTUSER username [ , acctname ] [ ;PASS=[ password ] ] [ ;CAP=[ capabilitylist ] ]  
[ ;MAXPRI=[ subqueueuname ] ] [ ;LOCATTR=[ localattribute ] ]  
[ ;HOME=[ homegroupname ] ] [ ;USERPASS=[ { REQ } ] [ , EXPIRED ] ] (1)
```

(1) The USERPASS parameter is only available if the HP Security Monitor has been installed.

Parameters

username The name assigned to the user for a logon account.

password The new password to be assigned to the user. When you enter
“;PASS=Return”, any existing password is removed. If you omit this
parameter entirely, the password remains unchanged.

capabilitylist The list of capabilities, separated by commas, permitted to this user. The
capabilities assigned to the user cannot exceed those assigned to the account.
Each capability is denoted by a two-letter mnemonic as follows:

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

The default is SF, ND, IA, and BA. CV automatically gives the user UV capability.

*subqueue***name** The name of the highest priority subqueue that may be requested for any job or session initiated by the user. This parameter is specified as AS, BS, CS, DS, or ES, but cannot be greater than that specified with the NEWACCT or ALTACCT commands. The default is CS. The *subqueue***name** defined for the user is checked against the *subqueue***name** defined for the account at logon, and the lower priority of the two is used as the maximum priority for the job or session. If you make a request for priority when you log on, it is checked against the *subqueue***name** defined for you as the user, and you are granted the lower of the two values.

Caution

Processes capable of executing in the AS or BS subqueues can deadlock the system. By assigning non-priority processes to these subqueues, you may prevent critical system processes from executing. Exercise extreme care when assigning processes to the AS or BS subqueue.

localattribute Defined at the installation site, this is an arbitrary double-word bitmap used to further classify users. Although it is not part of standard MPE V/E security provisions, programmers may define it (through the WHO intrinsic) to enhance the security of their own programs. The bitmap for the user's local attributes must be a subset of the bitmap for the account's local attributes. The ALTUSER command checks the local attributes of the user with those of the account. The default is double-word 0 (null).

homegroup-name The name of an existing group to be assigned as the home group for this user. The first user established when an account is created is by default assigned PUB as the *homegroup*. Subsequent new users have by default no *homegroup* assigned. If no *homegroup* is assigned, the user must always specify an existing group when logging on.

REQ USERPASS=REQ specifies that the user must have a non-blank password. It is available only if the HP Security Monitor has been installed. Refer to the *HP Security Monitor User's Guide* (30392-90001) for more information.

OPT USERPASS=OPT specifies that this user may or may not have a password. It is available only if the HP Security Monitor has been installed.

EXPIRED The password expires immediately. The user cannot logon without selecting a new password. It is only available if the HP Security Monitor has been installed.

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** has no effect on this command. You must have account manager (AM) capability to use this command.

ALTUSER

Operation

The ALTUSER command allows the account manager to change the password, capabilities, processing subqueue, security checking, and home group currently defined for a user. More than one of these attributes may be changed at a time by entering multiple keyword parameters on a single command line, using the “;” delimiter.

To change an attribute, enter the keyword and its new value. If you enter the keyword and no corresponding parameter, the default value is assigned. When an entire keyword parameter group is omitted from the ALTUSER command, the corresponding value for the user remains unchanged. When a keyword is included, but the corresponding parameter is omitted (as in ;PASS=), a default value is assigned as follows:

Parameter	Default Values
<i>password</i>	Null (no password)
<i>capabilitylist</i>	SF,ND,IA,BA (provided these capabilities have been specified for the account)
<i>subqueuname</i>	CS
<i>localattribute</i>	Null (double-word 0)
<i>homegroupname</i>	The first user established when the account was created is assigned PUB as home. Subsequent users are not assigned a home group. If you were not assigned a home group, you must specify an existing group when initiating a job or a session.

When a parameter is modified with the ALTUSER command, it is immediately registered in the directory. However, it does not affect users who are currently logged on to the system. They are affected the next time they logon to the same user name and account. For this reason, you should warn users in advance of the intended changes.

You should avoid changing the *capabilitylist* or *homegroupname* of the user **MANAGER.SYS**. SM capability cannot be taken away from **MANAGER.SYS**.

If more than one user within an account is assigned AM capability, the capability can be removed from all but one user assigned it. The last user retains the AM capability.

Examples

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

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

To alter two attributes, *password* and *subqueue name*, for user JONES enter:

```
:ALTUSER JONES;PASS=JJ;MAXPRI=DS
```

Additional Discussion

HP Security Monitor User's Guide (30392-90001)



ALTVSET

Modifies the definitions of volume sets.

Syntax

ALTVSET *vsname*

[;ADDCLASS=*vcname:vsname*[, *vsname*[, ...]]]

[;EXPANDCLASS=*vcname:vsname*[, *vsname*[, ...]]]

[;EXPANDSET=*vsname:type*[, *vsname:type*[, ...]]]

Parameters

<i>vsname</i>	The existing volume set name. MPE V/E implicitly references <i>vsname</i> as <i>vsname.groupname.acctname</i> , where <i>groupname</i> and <i>acctname</i> are the logon group and account.
<i>vcname:vsname</i>	The volume class name, consisting of up to eight alphanumeric characters, beginning with an alphabetic character. Each volume class definition consists of a volume class name and an associated list of volume names, <i>vsname</i> . The list of volume names must be a subset of the volumes composing the volume set. One of the volume names must be that of the volume set's master volume.
ADDCLASS	Directs MPE V/E to add a volume class and an associated list of volume names to an existing volume set. The list of volume names must be a subset of the volumes in the volume set. One of the volume names must be that of the set's master volume.
EXPANDCLASS	Directs MPE V/E to add the names of new members in the volume class to an existing volume set. These new members can only be members of the parent volume set.
EXPANDSET	Directs MPE V/E to add the names of new members, and the types of devices required to accommodate them, to an existing volume set. Total members in a volume set may not exceed eight. You must specify the device type with the name of each new member. The following types of disk drives may be specified:

HP 7902	HP 7920	HP 7936	HP 7959B
HP 7905	HP 7925	HP 7937	HP 7961B
HP 7906	HP 7933	HP 7957	HP 7962B
HP 7911	HP 7935	HP 7957B	HP 7963B
HP 7912	HP 7945	HP 7958	
HP 7914	HP 9895	HP 7958B	

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have create volumes (CV) capability to use this command.

Operation

The **ALTVSET** command is used to modify volume set definitions. Volume class names may be added, and the number of member volumes may be increased. The overall size of the volume set, however, cannot be decreased by deleting member volumes. **ALTVSET** does not alter a volume set if it is in use, that is, if an explicit or implicit **MOUNT** command has been issued.

Example

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

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

Additional Discussion

Refer to the discussion of “Private Volumes” in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

ASSOCIATE

Gives system operator control of a device to a user.

Syntax

ASSOCIATE *devclass*

Parameters

devclass The name of a logical device class configured during SYSDUMP.

Use

You may issue this command from a session, program, or in **BREAK**. It may not be used from a job. Pressing **BREAK** has no effect on this command.

Operation

This command links a device class, such as LP, to an individual user on the system. The user may then execute any valid operator command for a device in the device class and receive the status messages for the devices in that device class on \$STDLIST. For example, a remote printer may be associated with a terminal, so that messages concerning the printer go to the terminal, not the system console.

Before a user can be associated, the system manager must run a utility program (the version of ASOCTBL5.PUB.SYS that matches your operating system) in order to create a device class/user association table. This table defines which users may be associated with which device classes. At any given time, only one user may be associated with a given device class. If the device belongs in several device classes, only one of those device classes may be associated.

The operator commands that can be made available to users through the **ASSOCIATE** command are:

ABORTIO	OUTFENCE
ACCEPT	MPLINE
ALTSPoolFILE	REFUSE
DELETESPoolFILE	REPLY
DISCRPS	RESUMESPool
DOWN	SHOWCOM
DOWNLOAD	SHUTQ
FOREIGN	STARTSPool
GIVE	STOPSPool
HEADON	SUSPENDSPool
HEADOFF	TAKE
OPENQ	UP

Both the system supervisor and the user may **DISASSOCIATE** a user from a device. In addition, a user implicitly disassociates a device when logging off.

Example

To become the controller of the device class “TAPE”, enter:

:ASSOCIATE TAPE

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005).

AUTOALLOCATE

Disables or enables the automatic allocation and deallocation of programs.

Syntax

AUTOALLOCATE { OFF
ON }

Parameters

- OFF** Turns off the automatic allocation or deallocation of programs. Also unloads all programs already loaded by this feature if they are not being used.
- ON** Starts the automatic allocation or deallocation of programs.

Use

You may issue this command from a session, job, program or in **BREAK** mode. Pressing **BREAK** has no effect on this command. You must have system supervisor (OP) capability to use this command.

Operation

This feature manages the automatic allocation and deallocation of programs in such a way that the process is transparent to the user. It reduces system overhead caused by loading programs for execution. It also increases the average response time of the **RUN** command, although it does not make programs execute faster.

As a program is run, it is automatically allocated (see the **ALLOCATE** command). When the program terminates, it stays in the autoallocated state. The next time the program is run, much less time is needed to load the program for execution. Programs stay in the autoallocated state after they are run until a resource that the program holds is needed. When this occurs, the program that has not executed for the longest amount of time is automatically deallocated and the resource it holds are released.

Note Since temporary program files and program files on private volumes cannot be allocated by the **ALLOCATE** command, they cannot be autoallocated either.



Examples

To start the automatic allocation and deallocation of programs, enter:

:AUTOALLOCATE ON

To stop the automatic allocation and deallocation of programs, enter:

:AUTOALLOCATE OFF

BASIC

Interprets a BASIC program. BASIC is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
BASIC [ commandfile ] [ , [ inputfile ] [ , listfile ] ]
```

Parameters

<i>commandfile</i>	Actual file designator of the source file or device from which BASIC commands and statements are input. This can be any ASCII input file. The formal file designator is BASCOM . The default is \$STDINX .
<i>inputfile</i>	Actual file designator of the file containing data input for a BASIC program. This can be any ASCII input file. The formal file designator is BASIN . The default is \$STDINX .
<i>listfile</i>	Actual file designator of the destination file for the program listing and output. This can be any ASCII output file. The formal file designator is BASLIST . The default is \$STDLIST .

Note



The formal file designators (**BASCOM**, **BASIN**, and **BASLIST**) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the **FILE** command.

Use

You may issue this command from a session or job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **BASIC** command is generally used for online programming in BASIC, but you can also use it to interpret BASIC programs submitted in batch mode. In batch mode, the **BASIC >EOD** command is required after any data following the **BASIC >RUN** command, or after the **>RUN** command itself if there is no data.

BASIC

Examples

To enter commands and data from your standard input device, with program listing and output transmitted to the standard output device, enter:

:BASIC

You may also submit commands and data to the BASIC interpreter through input files that you have stored on disk. Files you have created using the “Editor” subsystem must be kept with the **UNN** (unnumbered) option of the subsystem **KEEP** command. In the following example, BASIC interpreter commands and statements are submitted from the command file **MYCOMDS**. The data that the program uses is stored in the input file **MYDATA**. The program listing and output are written to the file **MYLIST**.

:BASIC MYCOMDS,MYDATA,MYLIST

Additional Discussion

BASIC/3000 Interpreter Manual (32103-90001)

MPE Segmenter Reference Manual (30000-90011)

BASICGO

Compiles, prepares, and executes a BASIC program. BASIC is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

BASICGO [*commandfile*] [,*listfile*]

Parameters

<i>commandfile</i>	Actual file designator of the input file from which the BASIC compiler commands are read. This can be any ASCII input file. The formal file designator is BSCTEXT. The default is \$STDINX.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is BSCLIST. The default is \$STDLIST.

Note



The formal file designators (BSCTEXT and BSCLIST) used in this command cannot be backreferenced as actual file designators in the command parameter list. Refer to the discussion on “Implicit FILE Commands for Subsystems” under the FILE command for further information.

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

Operation

The BASICGO command compiles, prepares, and executes a program from a “fastsave” file created by the BASIC interpreter. This enables the program to run faster than it would if it were executed by the interpreter.

To save the program after it is written, use the BASIC interpreter command **SAVE filename,FAST**. The program can then be compiled, prepared, and executed with the BASICGO command. You must specify the **FAST** option to compile the program.

BASICGO

Example

To compile, prepare, and execute the BASIC program MYPROG, enter:

<u>:BASICGO</u>	** Calls the BASIC compiler. **
<u>\$CONTROL USLINIT</u>	** Initializes USL file. **
<u>\$COMPILE MYPROG</u>	** Compiler program MYPROG. **
<u>\$EXIT</u>	** Exits from compiler. **

Additional Discussion

BASIC Compiler Reference Manual (32103-90001)

MPE Segmenter Reference Manual (30000-90011)

BASICOMP

Compiles a BASIC program. BASIC is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
BASICOMP [ commandfile ] [ , [ uslfile ] [ , listfile ] ]
```

Parameters

<i>commandfile</i>	Actual file designator of the input file from which the compiler commands are read. This can be any ASCII input file. The formal file designator is BSCTEXT. The default is \$STDINX.
<i>uslfile</i>	Actual file designator of the user subprogram library (USL) file to which the object code is written, which can be any binary output file with a file code of USL or 1024. Its formal file designator is BSCUSL. If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS. If entered, this parameter specifies that the file was created in one of four ways: ■ By using the SAVE command to save the default USL file \$OLDPASS, created by a previous compilation. ■ By building the USL file with the Segmenter command -BUILDUSL (refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011)). ■ By creating a new USL file with the BUILD command and a file code of USL or 1024. ■ By specifying a nonexistent <i>uslfile</i> parameter, thereby creating a permanent file of the correct size and type.
<i>listfile</i>	Actual file designator of the file on which the program listing is written. This can be any ASCII output file. The formal designator is BSCLIST. The default is \$STDLIST.

Note



The formal file designators (BSCTEXT, BSCUSL, and BSCLIST) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on "Implicit FILE Commands for Subsystems" under the FILE command.

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

BASICOMP

Operation

The **BASICOMP** command compiles a program from a “fastsave” file generated by the BASIC interpreter. If you do not specify a USL file, the BASIC compiler stores the object code in the default system-defined temporary file **\$OLDPASS**, as shown in the second example, below. You may, however, build a USL file in the permanent file domain, then direct the BASIC compiler to store the object code in this file by naming the USL file in the **BASICOMP** command line. The first example, below, does this.

Example

To compile the BASIC program **MYPROG** onto the USL file named **OBJECT**, enter:

```
:BUILD OBJECT;CODE=USL    ** Builds USL file. **  
:BASICOMP,OBJECT         ** Calls BASIC compiler; specifies USL **  
                           ** file named OBJECT.**  
$CONTROL USLINIT         ** Initializes USL file. **  
$COMPILE MYPROG          ** Compiles fastsave program namedMYPROG. **  
$EXIT                   ** Exits from compiler. **
```

If you do not choose to build a USL file, the **BASICOMP** command compiles your program, storing the object code in the default USL file **\$OLDPASS**.

```
:BASICOMP    ** Runs BASIC compiler, accepting commands from **  
              ** $STDINX, and specifying $OLDPASS for the USL **  
              ** file and $STDLIST for listing output.**  
  
$COMPILE MYRUN    ** Compiles from fastsave file named MYRUN into **  
                  ** a USL file named $OLDPASS.**  
  
$EXIT           ** Exits from BASIC compiler. **
```

If you now want to run your program, use the **PREPRUN** command:

```
:PREPRUN $OLDPASS ** Prepares and runs program. **
```

Additional Discussion

BASIC Compiler Reference Manual (32103-90001)

MPE Segmenter Reference Manual (30000-90011)

BASICPREP

Compiles and prepares a BASIC program. BASIC is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
BASICPREP [ commandfile ] [ , [ progfile ] [ , listfile ] ]
```

Parameters

<i>commandfile</i>	Actual file designator of the input file from which the compiler commands are read. This can be any ASCII file. The formal file designator is BSCTEXT. The default is \$STDINX.
<i>progfile</i>	Actual file designator of the program file on which the prepared program segments are written. You get the best results by omitting the <i>progfile</i> parameter. When the <i>progfile</i> parameter is omitted, the Segmenter creates the program file, which resides in the temporary file domain as \$OLDPASS. If you do create your own program file, you must do so in one of two ways: ■ By using the BUILD command, and specifying a file code of 1029 or PROG and a <i>numextents</i> value of 1. This file is then used by the PREP command. ■ By specifying a nonexistent file in the <i>progfile</i> parameter, in which case a temporary job file of the correct size and type is created.
<i>listfile</i>	Actual file designator of the file to which the listing is written. This can be any ASCII output file. The formal file designator is BSCLIST. The default is \$STDLIST.

Note



The formal file designators (BSCTEXT and BSCLIST) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the FILE command.

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

Operation

The BASICPREP command compiles and prepares a program for execution from a “fastsave” file generated by the BASIC interpreter. If you omit the *progfile* parameter, the prepared program segments are stored in the system-defined temporary file \$OLDPASS. If you want to save the prepared program in a file other than \$OLDPASS, you may either create a file and specify its filename on the BASICPREP command line, or specify a non-existent *progfile*.

A program that you have compiled and prepared with the BASICPREP command may be executed with the RUN command.

BASICPREP

Example

To compile and prepare a program named MYPROG from the BASIC “fastsave” file named MYCOMDS, with listing directed to the standard list device, enter:

:BASICPREP,MYCOMDS

The file MYPROG is an ASCII file that contains the following BASIC compiler commands:

<u>\$CONTROL USLINIT,SOURCE</u>	** Initializes USL file and **
	** lists program. **
<u>\$COMPILE MYPROG</u>	** Compiles fastsave program. **
<u>\$EXIT</u>	** Exits from compiler. **

Additional Discussion

BASIC Compiler Reference Manual (32103-90001)

BBASIC

Interprets an HP Business BASIC program. HP Business BASIC is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
BBASIC [ commandfile ] [ , [ inputfile ] [ , listfile ] ]
```

Parameters

<i>commandfile</i>	Actual file designator of the sourcefile or device from which BBASIC commands and statements are input. This can be any ASCII input file. The formal file designator is BASCOM . The default is \$STDINX .
<i>inputfile</i>	Actual file designator of the file containing data input for a BBASIC program. This can be any ASCII input file. The formal file designator is BASIN . The default is \$STDINX .
<i>outfile</i>	Actual file designator of the destination file for the program listing and output. This can be any ASCII output file. The formal file designator is BASOUT . The default is \$STDLIST .

Note



The formal file designators (**BASCOM**, **BASIN**, and **BASOUT**) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the **FILE** command.

Use

You may issue this command from a session or job. You may not use it from a program or in **BREAK**. Pressing **(BREAK)** suspends execution.

Operation

The **BBASIC** command is generally used for online programming in **BBASIC**, but it can also be used to interpret **BBASIC** programs submitted in batch mode. In batch mode, the **BBASIC >EOD** command is required after any data following the **BBASIC >RUN** command, or after the **>RUN** command itself if there is no data. Business BASIC has its own online **HELP** facility.

BBASIC

Example

To enter commands and data from your standard input device, with program listing and output transmitted to the standard output device, use:

:BBASIC

You may also submit commands and data to the BBASIC interpreter through input files that you have stored on disk. Files created using the Editor must be kept with the UNN (unnumbered) option of the Editor's KEEP command. In this example, BBASIC interpreter commands and statements are submitted from the command file MYCOMDS. The data that the program uses is stored in the input file MYDATA. The program listing and output are written to the file MYLIST.

:BBASIC MYCOMDS,MYDATA,MYLIST

Additional Discussion

HP Business BASIC Reference Manual (32115-90001)

MPE Segmenter Reference Manual (30000-90011)

BBASICGO

Compiles, prepares, and executes an HP Business BASIC program. HP Business BASIC is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

BBASICGO *infile* [, *listfile*]



Parameters

<i>infile</i>	Actual file designator of the BSAVE file containing the BBASIC program to be compiled. The formal file designator is BBCIN.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is BBCLIST. The default is \$STDLIST.

Note



The formal file designators (BBCIN, and BBCLIST) used in this command cannot be backreferenced as actual file designators in the command parameter list. Refer to the discussion on “Implicit FILE Commands for Subsystems” under the FILE command in this chapter.

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

Operation

The BASICGO command compiles, prepares, and executes a program from a BSAVE file created by the BBASIC interpreter. This enables the program to run faster than it would if it were executed by the interpreter.

You may create a BSAVE program file within the BBASIC interpreter after it is saved by using the BBASIC interpreter command **>SAVE filename**. The program can then be compiled, prepared, and executed with the BBASICGO command.

BBASICGO

Example

To compile, prepare, and execute the BBASIC program MYPROG and send the listing to the disk file LISTFL, enter:

```
:BBASICGO MYPROG,LISTFL
```

Additional Discussion

HP Business BASIC Reference Manual (32115-90001)

Basic Compiler Reference Manual (32103-90001)

MPE Segmenter Reference Manual (30000-90011)

BBASICOMP

Compiles an HP Business BASIC program. HP Business BASIC is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
BBASICOMP infile[ , [ uslfile ] [ , listfile ] ]
```

Parameters

<i>infile</i>	Actual file designator of the BSAVE file containing the BBASIC program to be compiled. This can be any ASCII input file. The formal file designator is BBCIN .
<i>uslfile</i>	Actual file designator of the user subprogram library (USL) file on which the object program is written, which can be any binary output file with a file code of USL or 1024. Its formal file designator is BBSCUSL . If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS . If entered, this parameter specifies that the file was created in one of four ways: ■ By using the SAVE command to save the default USL file \$OLDPASS created by a previous compilation. ■ By building the USL file with the Segmenter command -BUILDUSL. Refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011). ■ By creating a new USL file with the BUILD command and specifying a file code of USL or 1024. ■ By specifying a nonexistent <i>uslfile</i> parameter, thereby creating a permanent file of the correct size and type.
<i>listfile</i>	Actual file designator of the file on which the program listing is written. This can be any ASCII output file. The formal designator is BBCLIST . The default is \$STDLIST .

Note



The formal file designators (**BBCIN**, **BBCUSL**, and **BBCLIST**) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the **FILE** command.

Use

You may issue this command from a session or job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

BBASICOMP

Operation

The BBASICOMP command compiles a program from a BSAVE file generated by the BBASIC interpreter. This enables the program to run faster than it would if it were left in the form generated by the interpreter.

A BSAVE program file can be created from within the BBASIC interpreter after it is written, by using the BBASIC interpreter command >SAVE *filename*. The program may be compiled with the BBASICOMP command, then prepared with the command PREP, and executed with the command RUN.

Examples

To compile the BBASIC program MYPROG onto the USL file named OBJECT, enter:

```
:BBASICOMP MYPROG,OBJECT
```

If you do not choose to build a USL file, the BASICOMP command compiles your program, storing the object code in the default USL file \$OLDPASS.

```
:BBASICOMP MYPROG  ** Runs BASIC compiler and accepts MYPROG**  
                    ** as a BSAVE program file. $OLDPASS is **  
                    ** the default USL file, and $STDLIST is the **  
                    ** default output listing.**
```

If you now want to run your program, use the PREPRUN command:

```
:PREPRUN $OLDPASS  ** Prepares and runs program. **
```

Additional Discussion

HP Business BASIC Reference Manual (32115-90001)

Basic Compiler Reference Manual (32103-90001)

MPE Segmenter Reference Manual (30000-90011)

BBASICPREP

Compiles and prepares an HP Business BASIC program. HP Business BASIC is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
BBASICPREP infile [ , [ progfile ] [ , listfile ] ]
```

Parameters

<i>infile</i>	Actual file designator of the BSAVE file containing the BBASIC program to be compiled. The formal file designator is BBCIN.
<i>progfile</i>	Actual file designator of the program file on which the prepared program segments are written. You get the best results by omitting the <i>progfile</i> parameter. If the <i>progfile</i> parameter is omitted, the Segmenter creates the program file, which resides in the temporary file domain as \$OLDPASS. If you do create your own program file, you must do so in one of two ways: ■ By using the BUILD command and specifying a file code of 1029 or PROG and a <i>numextents</i> value of 1. The PREP command then uses this file. ■ By specifying a nonexistent file in the <i>progfile</i> parameter, in which case a temporary job file of the correct size and type is created.
<i>listfile</i>	Actual file designator of the file on which program the listing is written. This can be any ASCII output file. The formal file designator is BBCLIST. The default is \$STDLIST.

Note



The formal file designators (BBCIN and BBCLIST) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the FILE command.

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

Operation

The BBASICPREP command compiles and prepares a program from a BSAVE file generated by the BBASIC interpreter. If you omit the *progfile* parameter, the prepared program segments are stored in the system-defined temporary file \$OLDPASS. If you want to save the prepared program in a file other than \$OLDPASS, you may either create a file and specify its filename on the BBASICPREP command line, or specify a non-existent *progfile*.

BBASICPREP

A BSAVE program file can be created from within the BBASIC interpreter after it is written, by using the BBASIC interpreter command **>SAVE *filename***. The program may be compiled with the **BBASICOMP** command, then prepared with the command **PREP**, and executed with the command **RUN**.

Example

To compile and prepare a program named **MYPROG** from the BBASIC BSAVE file named **MYCOMDS**, and send the listing to the standard list device, enter:

```
:BBASICPREP MYCOMDS,MYPROG
```

Additional Discussion

HP Business BASIC Reference Manual (32115-90001)

Basic Compiler Reference Manual (32103-90001)

MPE Segmenter Reference Manual (30000-90011)

BREAKJOB

Suspends an executing job.

Syntax

```
BREAKJOB #Jnnn
```

Parameters

#Jnnn A job number.

Use

You may issue this command from a session, job, or program, and in **BREAK**. Pressing **BREAK** has no effect on this command. You can execute it only from the console *unless* it is distributed to users with the **ALLOW** command or if **JOBSECURITY** is set to **LOW**.

Operation

The system operator can use the **BREAKJOB** command to suspend any executing job, including spooled and streamed jobs. A job using a critical system resource is not suspended until it releases the resource.

When you issue **BREAKJOB** for a job that controls a nonsharable device, a console message is displayed listing the device(s) that the job controls. As many as ten devices may be listed. You may then decide whether the job should be allowed to run until it releases the device(s), or whether it should be aborted.

All commands that normally affect executing jobs, such as **ABORTJOB**, operate on suspended jobs. The **SHOWJOB** command, which lists all jobs, displays "SUSP" for those in the suspended state. To list suspended jobs only, enter "**SHOWJOB SUSP**".

Examples

To suspend job #68, enter:

```
:BREAKJOB #J68
```

To display suspended jobs, enter:

```
:SHOWJOB SUSP
JOBNUM  STATE  INPRI  JIN  JLIST  INTRODUCED  JOB  NAME
#68     SUSP           105   LP   WED. 7:56AM TEST,USER.ACCOUNT
```

BUILD

Creates and immediately allocates a new empty file on disk.

Syntax

$$\text{BUILD } \textit{filereference} \left[\begin{array}{l} \text{;REC= [} \textit{recsize} \text{] } \left[\begin{array}{l} \text{, [} \textit{blockfactor} \text{] } \left[\begin{array}{l} \left[\left\{ \begin{array}{c} \text{F} \\ \text{U} \\ \text{V} \end{array} \right\} \right] \left[\left\{ \begin{array}{c} \text{BINARY} \\ \text{ASCII} \end{array} \right\} \right] \end{array} \right] \end{array} \right] \\ \left[\left\{ \begin{array}{c} \text{;CCTL} \\ \text{;NOCCTL} \end{array} \right\} \right] \left[\text{;TEMP} \right] \left[\text{;DEV= [[} \textit{dsdevice} \text{] } \# \text{ [} \textit{device} \text{] } \right] \left[\text{;CODE= } \textit{filecode} \right] \\ \left[\text{;DISC= [} \textit{numrec} \text{] [, [} \textit{numextents} \text{] [, [} \textit{initialloc} \text{]]] } \left[\left\{ \begin{array}{c} \text{;RIO} \\ \text{;NORIO} \end{array} \right\} \right] \left[\left\{ \begin{array}{c} \text{;MSG} \\ \text{;CIR} \end{array} \right\} \right] \end{array} \right]$$

Parameters

filereference Actual name of the file to be created, in the following format:

filename[/*lockword*][*.groupname*[*.acctname*]]

Names must contain from one to eight alphanumeric characters, beginning with an alphabetic character. If *acctname* is specified, it must be that of your logon account (you cannot create a file in another account). The default *groupname* and *acctname* are the logon group and account.

recsize The record size of a new file. A positive number specifies the size in words while a negative number specifies the size in bytes. For fixed length files, *recsize* is the logical record size. For undefined length files, *recsize* is the maximum record size. For variable length files using a blocking factor of 1, *recsize* is the maximum logical record size. If *blockfactor* is a value other than 1, *recsize* is used to calculate the maximum logical and physical record size.

Since records always begin on word boundaries, the record size is rounded up to the nearest word boundary to calculate block size. If the file format is binary or variable-length ASCII, odd byte lengths are rounded up and the extra byte is not available for data. For example, in a fixed-length ASCII file with *recsize* equal to 11 bytes, only 11 bytes are available for data in each logical record. To determine the block size, *recsize* is rounded up to 12 bytes. If the file format is binary, all 12 bytes, or 6 words, are available to store data.

The default record size is the configured physical record width of the associated device. Disk files have a default record size of 128 words unless you specify a device other than DISC using the ;DEV= parameter.

<i>blockfactor</i>	The number of logical records per physical block in a new file. The default is calculated by dividing the specified <i>recsize</i> into the configured block size; this value is rounded down to an integer that is never less than 1. For variable-length record files, <i>blockfactor</i> and <i>recsize</i> are used to calculate the maximum logical and physical record size. The <i>blockfactor</i> is then set to 1. For files containing records of undefined length, the <i>blockfactor</i> is ignored. The maximum size of <i>blockfactor</i> is 255.
F,U,orV	Defines the length of the records of the file. A file may contain records of fixed length (F), records of undefined length (U), or variable-length records (V). For disk files, the default is F.
BINARY or ASCII	Indicates the type of records the file contains. BINARY indicates records in binary code and is the default. ASCII indicates records coded in ASCII.
CCTL or NOCCTL	Indicates whether or not carriage control characters are supplied along with data written to an ASCII file. CCTL indicates carriage control characters accompany the data; NOCCTL indicates carriage control characters are not specified. The default is NOCCTL.
TEMP	Indicates that the file is created as a temporary jobfile and is saved in the temporary file domain of the job/session when it is closed. Refer to the <i>MPE File System Reference Manual</i> (30000-90236) for more information on file domains. The default is that a permanent file is created.
<i>dsdevice</i>	The device class name or logical device number used to open a communications link to a remote computer that contains the source file. The default is the local system, or the computer on which the transfer request originates. A “#” symbol is the delimiter between the filename of the remote computer and the filename of the remote device.
<i>device</i>	Specifies either the device class or logical device number on which the file is to reside. A device class name, <i>devclass</i> , consists of up to eight alphanumeric characters beginning with an alphabetic character, such as DISC or PVDISC1. When you specify <i>devclass</i> , the file is allocated to any available device in that class. If you are opening a file destined for a private volume, you must specify a device class which includes the drives upon which the home volume set is mounted. The file is then allocated to any of the volumes in the home volume set that fall within that device class. The logical device number, <i>ldev</i> , consists of a number from one to three digits long, specifying a particular device. The default is the device class name DISC.
<i>filecode</i>	A code indicating a specially formatted file. This code is recorded in the file label and is available to processes accessing the file through the FFILEINFO or FGETINFO intrinsic. Although any user can specify a positive integer ranging from 0 to 32767 or a mnemonic name for this parameter, certain reserved integers and mnemonics have meanings defined by the system, as follows:

BUILD

Integer	Mnemonic	Meaning
1024	USL	User Subprogram Library
1025	BASD	BASIC Data
1026	BASP	BASIC Program
1027	BASFP	BASIC Fast Program
1028	RL	Relocatable Library
1029	PROG	Program file
1030	NMPRG	Native Mode Program file
1031	SL	Segmented Library
1032	NMSL	Native Mode Executable Library file
1033	NMRL	Native Mode Relocatable Library file
1035	VFORM	View Form file
1036	VFAST	View Fast Forms file
1037	VREF	View Reformat file
1040	XLSAV	Cross Loader ASCII file (SAVE)
1041	XLBIN	Cross Loader Relocated Binary file
1042	XLDSP	Cross Loader ASCII file (DISPLAY)
1050	EDITQ	Edit Quick file
1051	EDTCQ	Edit KEEPQ file (COBOL)
1052	EDTCT	Edit TEXT file (COBOL)
1054	TDPDT	TDP Diary file
1055	TDPQM	TDP Proof Marked QMARKED
1056	TDPP	TDP Proof marked non-COBOL file
1057	TDPCP	TDP Proof Marked COBOL file
1058	TDPQ	TDP Workfile
1059	TDPXQ	TDP Workfile (COBOL)
1060	RJEPN	RJE Punch file
1070	QPROC	QUERY Procedure file
1080	KSAMK	KSAM Key file
1083	GRAPH	GRAPH Specification file
1084	SD	Self-describing file
1090	LOG	User Logging logfile
1100	WDOC	HPWORD Document
1101	WDICT	HPWORD Hyphenation dictionary
1102	WCONF	HPWORD Configuration file
1103	W2601	HPWORD Attended Printer Environment
1110	PCELL	IFS/3000 Character Cell file
1111	PFORM	IFS/3000 Form file
1112	PENV	IFS/3000 Environment file
1113	PCCMP	IFS/3000 Compiled Character Cell file
1114	RASTR	Graphics Image in RASTR Format
1130	OPTLF	OPT/3000 logfile
1131	TEPES	TEPE/3000 Script file
1132	TEPEL	TEPE/3000 logfile
1133	SAMPL	APS/3000 logfile

Integer	Mnemonic	Meaning
1139	MPEDL	MPEDCP/DRP logfile
1140	TSR	HPToolset Root file
1141	TSR	HPToolset Data file
1145	DRAW	Drawing File for HPDRAW
1146	FIG	Figure File for HPDRAW
1147	FONT	Reserved
1148	COLOR	Reserved
1149	D48	Reserved
1152	SLATE	Compressed SLATE file
1153	SLATW	Expanded SLATE workfile
1156	DSTOR	Store file for RAPID/3000 Utility DICTDBU
1157	TCODE	Code file for Transact/3000 Compiler
1158	RCODE	Code file for Report/3000 Compiler
1159	ICODE	Code file for Inform/3000 Compiler
1166	MDIST	HPDESK Distribution list
1167	MTEXT	HPDESK Text
1168	MARPA	ARPA Message file
1169	MARPD	ARPA Distribution list
1170	MCMND	HPDESK Abbreviated Commands file
1171	MFRM	Reserved
1172		Reserved
1173	MEFT	Reserved
1174	MCRPT	Reserved
1175	MSERL	Reserved
1176	UCSF	Reserved
1177	TTYPE	Term Type file
1178	TVFC	Term Vertical Format Control file
1192	NCONF	Network Configuration file
1193	NTRAC	Network Trace file
1194	NLOG	Network logfile
1195	MIDAS	Reserved
1211	ANODE	Reserved
1212	INODE	Reserved
1213	INVRT	Reserved
1214	EXCEP	Reserved
1215	TAXON	Reserved
1216	QUERF	Reserved
1217	DOCDR	Reserved
1226	VC	VC file
1227	DIF	DIF file
1228	LANGD	Language Definition file
1229	CHARD	Character Set Definition File
1230	MGCAT	Formatted Application Message Catalog
1236	BMAP	Reserved
1242	BDATA	Basic Data file
1243	BFORM	Basic Field Order File for VPLUS
1244	BSAVE	Basic Saved Program file

BUILD

Integer	Mnemonic	Meaning
1245	BCNFG	Config. File for Default Option Basic Program
1246	BKEY	Function Key Definition file
1258	PFSTA	Pathflow STATIC file
1259	PFDYN	Pathflow DYNAMIC file
1270	RTDCA	Revisable Form DCA Document*
1271	FFDCA	Final Form DCA Document*
1272	DIU	Document Interchange Unit File
1273	PDOC	HPWORD/150 Document
1275	DFI	DISOSS Filing Information file
1276	SRI	Search Result Information file
1401	CWPTX	Reserved
1421	MAP	HPMAP/3000 Map Specification file
1422	GAL	Reserved
1425	TTX	Reserved
1428	RDIC	HPBRW Dictionary file
1429	RSPEC	HPBRW Specification file
1430	RSPCF	HPBRW Specification file (frozen)
1431	REXEC	HPBRW Execution file
1432	RJOB	HPBRW Report Job file
1433	ROUTI	HPBRW Intermediate Report file
1434	ROUTD	Dictionary Output file
1435	PRINT	HPBRW Print file
1436	RCONF	HPBRW Configuration file
1441	PIF	Reserved
1461	NMOBJ	Native Mode Object file
1462	PASLB	PASCAL Source Library
3333		Reserved

* DCA = Document Content Architecture

The default is the unreserved file code of 0.

If you use 1090 (LOG) as your designated *filecode*, you may not get the number of records you specify in the ;DISC= parameter. Most files use the number of records specified in the ;DISC= parameter as the maximum limit; user logging uses this number as a minimum.

numrec The maximum number of logical records in a new file. The theoretical maximum for fixed and undefined length records is 2,147,483,647. Since this maximum is limited by block size, the actual maximum number of records per file is 2,097,120, which is based on 65535 sectors per extent and 32 extents. An approximate practical limit for *numrec* is 2,097,119 for files with variable and undefined length records, and 267,382,000 for fixed length files. The default is 1023.

Note

The file system uses these values to compute other characteristics of the file as well. Therefore, the values (or default values) specified on the BUILD command may be valid within their respective fields, but may cause overflow errors when internally needed file specifications are computed. Refer to the *MPE File System Reference Manual* (30000-90236) for a discussion on calculating file space.

<i>numextents</i>	Maximum number of disk extents. This is a value from 1 to 32. The default is 8.
<i>initialloc</i>	Number of extents to be initially allocated to the file at the time it is opened. This is a value from 1 to 32. The default is 1.
RIO or NORIO	RIO creates a relative I/O file, which is a special file access method primarily used by COBOLII programs. You can, however, access these files from programs written in any language. Specifying RIO implicitly changes the record length parameter to F, or fixed record length. The default, NORIO, creates a non-relative I/O file. RIO and NORIO specifications affect only the physical characteristics of the file. If NOBUF is specified in the FILE command, the file is accessed in non-RIO mode; otherwise RIO access is used with RIO files. Special operations on RIO files, such as replicating a RIO file, uses NOBUF access. Refer to the <i>MPE V Intrinsic Reference Manual</i> (32033-90007) for a discussion of relative I/O.
STD,MSG, or CIR	Defines the type of file. The default, STD, is a standard MPE V/E disk file. A MSG, or message, file allows communication between any set of processes in a first in, first out (FIFO) manner. Records are read from the start of the file and logically deleted and/or are appended to the end of the file. A CIR, or circular, file acts as a normal sequential file until full. When full, the first physical block is deleted when the next record is written, and remaining blocks are logically shifted to the front of the file. A circular file cannot be simultaneously accessed by readers and writers.

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** has no effect on this command.

BUILD

Operation

The BUILD command builds a new file on disk and immediately allocates space for it. If it is an ASCII file, the file space is initialized to blanks. If it is a binary file, the file space is initialized to zeros.

Unless the TEMP parameter is specified, the file is saved in the permanent file domain. To create a permanent file, you must have save files (SF) capability and SAVE access to the group to which the new file belongs. You can only build a file belonging to your logon account.

If specified, the ;DEV=" " parameter must be consistent with the group to which the new file belongs. If the group's home volume set is not mounted, BUILD implicitly generates a volume set mount request. Refer to the *MPE File System Reference Manual* (30000-90236) for more information on file domains.

The default characteristics of a file created with the BUILD command are: fixed length records of 128 words each, a blocking factor of 1, binary formatted, permanent file, a record limit of 1023, and a maximum of 8 extents with 1 extent initially allocated. This is equivalent to entering:

```
:BUILD filename;REC=128,1,F,BINARY;DEV=DISC;DISC=1023,8,1
```

Examples

The following example creates a permanent disk file named WORKFILE, which can reside on any disk. WORKFILE has fixed length records of 80 bytes each. The records are blocked 3 records per block (which is the *blockfactor*), and are written in ASCII code. The file has a maximum capacity of 2000 records divided into 10 extents with 2 extents allocated immediately.

```
:BUILD WORKFILE;REC=-80,3,F,ASCII;DISC=2000,10,2
```

The second example uses the BUILD command to create a permanent disk file, named VFILE, belonging to the logon (default) group that resides on the volume set specified by VCLASS1:

```
:BUILD VFILE;DISC=500,10,1;REC=-80;DEV=VCLASS1
```

(VCLASS1 is the name of the existing volume class.)

The following example uses the CODE= parameter to create a logging file called NEWDATA:

```
:BUILD NEWDATA;DISC=3000,1,1;CODE=LOG
```

Additional Discussion

MPE File System Reference Manual (30000-90236)

MPE V Intrinsic Reference Manual (32033-90007)

BYE

Ends an interactive session.

Syntax

BYE

Parameters

None.

Use

You may issue this command from a session. You may not use it from a job, program, or in BREAK. Pressing **BREAK** has no effect on this command.

Operation

The BYE command terminates a session and displays the CPU time used (in seconds), connect time (in minutes), the date and time, as follows:

```
CPU=48. CONNECT=35. FRI, JUL 14, 1989, 10:56 PM
```

If you enter a HELLO command before logging off, MPE V/E terminates your current session and immediately initiates a new one. If you are logged onto the computer with a telephone connection, and you hang up before terminating your session, MPE V/E issues a BYE command automatically.

Example

To terminate a session, enter:

```
:BYE
```

```
CPU=1. CONNECT=5. FRI, JUL 14, 1989, 12:20 PM
```

CACHECONTROL

Tunes the performance of disk caching. Disk caching software is not included in the MPE V/E fundamental operating system and must be purchased separately.

Syntax

$$\text{CACHECONTROL} \left\{ \begin{array}{l} \text{SEQUENTIAL}=\text{sequentn} \\ \text{;RANDOM}=\text{randn} \\ \text{;BLOCKWRITE}=\left\{ \begin{array}{l} \text{YES} \\ \text{NO} \end{array} \right\} \end{array} \right\} [; \dots]$$

Parameters

- sequentn* The number of sectors to be read from a sequential access disk file if the information is not found in the cache. The disk read stops at the extent boundaries and fetches at least the number of sectors requested. The number of sectors must be greater than 1, but less than 96 (inclusive). The default is 96 sectors.
- randn* The number of sectors to be read from a random-access disk file if the information is not found in the cache. The disk read stops at the extent boundaries and fetches at least the number of sectors requested. The number of the sectors must be greater than 1, but less than or equal to 96. The default is 16 sectors.
- BLOCKONWRITE** Specifies whether MPE V/E will block the process until the posting of cache buffers to the disk is complete. The **BLOCKONWRITE** must be equal to **YES** or **NO**. The default is **NO**.

Use

You may issue this command from a session, job, or program. You may not issue it in **BREAK**. Pressing **(BREAK)** has no effect on this command. You require system supervisor (OP) or system manager (SM) capability to use it.

Operation

The **CACHECONTROL** command tunes the performance of disk caching on your system. You must specify at least one keyword. Any combination of keywords can be used in a single command, but each can be used only once. They can be entered in any order.

When an I/O **READ** request fails to find data currently in memory (a **READ** “miss”), the requested portion of the disk must be fetched into memory. Disk caching uses the information about the file extent size and access type (random or sequential) in its fetching strategy to maximize the “hit” ratio; that is, subsequent **READ** requests may be satisfied by a disk domain fetched as a result of a “miss”.

The size of the domain fetched is limited by the extent size of the file. Within this limit, the size of the disk domain fetched equals the greater of the following:

- The requested size, or
- The largest multiple of the requested size (in integers) that is smaller than the value specified by the *sequentn* or *randn* parameters.

If the actual request is greater than the value for *sequentn* then caching fetches the actual size requested. If it is less than or equal to the value for *sequentn*, then caching uses the largest multiple of the requested size that is still less than the configured *sequentn* or *randn* value. A high hit ratio on READ requests helps maximize disk caching performance. A poor choice of values for either or both of the *sequentn* and *randn* parameters can substantially decrease the hit ratio.

When disk caching is enabled on your system, data residing in the disk domain is transferred from the portion of memory reserved as the disk cache, and not from the disk itself. This way, I/Os complete at processor speed, rather than at disk speed.

When the I/O request is a WRITE operation, physical posting to the disk occurs as a background task while the process continues. In this way, WRITE requests appear complete to the user before the disk has been updated. By specifying the BLOCKONWRITE=YES parameter, no user-generated WRITE requests appear as completed until they are actually posted to the disk. This guarantees that the information is on disk when the request completes. However, the YES option does eliminate one performance benefit that disk caching normally provides: WRITE operations are completed at disk access speed, rather than at processor speed.

Example

To set the fetch values for random-access files to 24-sector reads, and the value for sequential access files to 48, enter:

```
:CACHECONTROL SEQUENTIAL=48,RANDOM=24
```

In this example, the default value of BLOCKONWRITE, which is NO, is used.



CHANGELOG

Changes the user logging file without stopping or interrupting the logging process.

Syntax

```
CHANGELOG logid [ ;DEV=device ]
```

Parameters

<i>logid</i>	Name of the currently active use logging process. This name may contain from one to eight alphanumeric characters, beginning with an alphabetic character.
<i>device</i>	Name of the device on which the new logging file is to be created. The device may be any one of the following: DISC, TAPE, SDISC, or CTAPE. The default is DISC.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **(BREAK)** has no effect on this command. It is available only to the creator of the *logid* and requires system supervisor (OP) or user logging (LG) capability.

Operation

This command permits the user to change the active logging file without stopping the logging process with the LOG *logid*, STOP command. By specifying a device, you may switch the logging file from one device to another, regardless of the device on which the logging file was created. If you have enabled automatic logging with the ALTLOG or GETLOG command, however, the only device available for logging is the default, DISC.

If the CHANGELOG command is valid, the system writes a CHANGELOG record to the end of the current logging file and closes the file. It then opens a new logging file whose characteristics are identical to those of the preceding file and makes the new file permanent. If the system is unable to open a new file of the same size, it tries to open a new file half the size of the old file. If the system determines that the half-size file is too small to continue logging, user logging terminates.

If the system opens a new logfile, it immediately writes a CHANGELOG record to the new file. The CHANGELOG record posted to the old logging file contains the fully qualified identifier of the new logging file. A corresponding CHANGELOG record written to the new file contains the fully qualified identifier of the old logging file. CHANGELOG records also contain the device type of the logging file to which the CHANGELOG refers.

The following messages is displayed on the \$STDLIST to confirm a successful change:

```
Log file for logid AAA has been changed from A001.PUB.SYS to  
A002.PUB.SYS (ulogmsg 38)
```

CHANGELOG

If the new logging file is a serial file, a message appears on the console advising the system operator to mount the new log file:

```
Mount new {tape/cartridge tape/serial disk} volume for logid
(ulogmsg 40).
```

When a user logging file is full, the system terminates the logging process and displays an appropriate message.

However, by specifying the **AUTO** parameter in a **GETLOG** or **ALTLOG** command, you can enable an automatic **CHANGELOG** procedure and free yourself of the need to issue the **CHANGELOG** command manually.

In order to use **CHANGELOG** (manually or automatically), you must end the first user logging filename with the numeric characters "001" (for example, *fname001*). This establishes a naming convention that works in conjunction with the fileset number to generate sequential filenames independently. New filenames consist of the filename root (*fname*) plus the next sequential increment of the last three digits:

Current File	Next File
TEST001	TEST002
TEST002	TEST003
...	...
TEST998	TEST999
TEST999	TEST000

The logging process opens files and automatically names them with the next sequential number, up to a maximum of 999. Thereafter, the numbering sequence is reset to 000 and begins incrementing all over again.

Automatic logging with the **CHANGELOG** command is available only for disk files.

Note



The logging process specified by *logid* must be currently active. If the logging process is in another state, such as "RECOVERING", "STOP," or "INITIALIZING", or if the logging process has another **CHANGELOG** pending, the command terminates in an error state. The **ALTLOG** command permits you to change the log file for an inactive logging process. **ALTLOG**, however, does not provide a way to link log files into a set.

Additional Discussion

Refer to the discussion on "Optional Capability" in section III of the *MPE V Intrinsic Manual* (32033-90007) and the commands **LISTLOG** and **SHOWLOGSTATUS** in this chapter.

COBOL

Compiles a COBOL program on the COBOL 68 compiler. COBOL is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
COBOL [ textfile ] [ , [ uslfile ] [ , ] listfile ] [ , [ masterfile ] [ , newfile ] ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is COBTEXT. The default is \$STDIN.
<i>uslfile</i>	Actual file designator of the user subprogram library (USL) file to which the object program is written, which can be any binary output file with a file code of USL or 1024. The formal file designator is COBUSL. If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS. If entered, this parameter indicates that the file was created in one of four ways: ■ By saving, using the SAVE command to save the default USL file \$OLDPASS created during a previous compilation. ■ By building the USL with the Segmenter command -BUILDUSL. Refer the <i>MPE Segmenter Reference Manual</i> (30000-90011). ■ By creating a new USL file with the BUILD command and specifying a file code of USL or 1024. ■ By specifying a nonexistent <i>uslfile</i> parameter, thereby creating a permanent file of the correct size and type.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is COBLIST. The default is \$STDLIST.
<i>masterfile</i>	Actual file designator of the master file with which <i>textfile</i> is merged to produce a composite source. This can be any ASCII input file. The formal file designator is COBMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if the <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the file created by the <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. The formal file designator is COBNEW. The default is that no file is written.

Note



The formal file designators (COBTEXT, COBUSL, COBLIST, COBMAST, and COBNEW) used in this command cannot be backreferenced as literal file identifiers in the command parameter list. For further information, refer to the “Implicit FILE Commands for Subsystems” discussion of the FILE command.

Use

You may issue this command from a session or job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **COBOL** command compiles a COBOL source program into a USL file on disk. If you do not specify a source textfile, MPE V/E expects input from your standard input device. If you create the USL file prior to compilation, you must specify a file code of USL or 1024. If you do not specify a *listfile*, MPE V/E sends the program listing to your standard list device.

Example

To compile a COBOL program from a file named **SOURCE** into an object program stored in the USL file **\$OLDPASS** sending the listing to your standard list device, enter:

```
:BUILD OBJECT;CODE=USL  
:COBOL SOURCE,OBJECT
```

Additional Discussion

COBOL/3000 Reference Manual (32213-90001)

MPE Segmenter Reference Manual (30000-90011)

COBOLGO

Compiles, prepares, and executes a COBOL program on the COBOL 68 compiler. COBOL is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
COBOLGO [ textfile ] [ , [ listfile ] [ , [ masterfile ] [ , newfile ] ] ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is COBTEXT. The default is \$STDIN.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal designator is COBLIST. The default is \$STDLIST.
<i>masterfile</i>	Actual file designator of the master file with which <i>textfile</i> is merged to produce a composite source. This can be any ASCII input file. The formal file designator is COBMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the file created by the merger <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. The formal file designator is COBNEW. The default is that no file is written.

Note



The formal file designators used in this command (COBTEXT, COBLIST, COBMAST, and COBNEW) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the FILE command.

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

Operation

The COBOLGO command compiles, prepares, and executes a COBOL program. If you do not specify *textfile*, MPE V/E expects input from your standard input device. If you do not specify *listfile*, MPE V/E sends the listing to your standard list device.

The USL file created during compilation is a system-defined temporary file called \$NEWPASS, which is passed directly to the Segmenter. The Segmenter writes the prepared program to \$OLDPASS, which can then be executed.

Example

To compile, prepare, and execute a COBOL program entered from your standard input device, with the program listing sent to your standard list device, enter:

:COBOLGO

To compile, prepare, and execute a COBOL program from the disk file TEXTFL and send the program listing to the disk file LISTFL, enter:

:COBOLGO TEXTFL,LISTFL

Additional Discussion

COBOL/3000 Reference Manual (32213-90001)

MPE Segmenter Reference Manual (30000-90011)

COBOLPREP

Compiles and prepares a COBOL program on the COBOL 68 compiler. COBOL is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
COBOLPREP [ textfile ] [ , [ progfile ] [ , [ listf ] [ , [ masterf ] [ , newfile ] ] ] ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is COBTEXT. The default is \$STDIN.
<i>progfile</i>	Actual file designator of the program file to which the prepared program segments are written. You get the best results by omitting the <i>progfile</i> parameter. When you omit <i>progfile</i>, the Segmenter creates the program file which resides in the temporary file domain as \$OLDPASS. The formal file designator is COBPROG. If you wish to create your own program file, you must do so in one of two ways: ■ By using the BUILD command, and specifying a file code of 1029 or PROG, and a <i>numextents</i> value of 1. This file is then used by the PREP command. ■ By specifying a nonexistent file in the <i>progfile</i> parameter, thereby creating a job file of the correct size and type.
<i>listf(ile)</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is COBLIST. The default is \$STDLIST.
<i>masterf(ile)</i>	Actual file designator of the master file with which <i>textfile</i> is merged to produce a composite source. This can be any ASCII input file. The formal file designator is COBMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the merged <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. The formal file designator is COBNEW. The default is that no file is written.

Note



The formal file designators (COBTEXT, COBLIST, COBMAST, and COBNEW) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the FILE command.

Use

You may issue this command from a session or job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **COBOLPREP** command compiles and prepares a COBOL program creating a program file on disk. If you do not specify *textfile*, MPE V/E expects the source program input from your standard input device. If you do not specify *listfile*, MPE V/E sends the output to your standard list device.

You get the best results by allowing the Segmenter to create your program file. You can, however, create your own permanent program file with the **BUILD** command. When you do, you must specify a file code of **PROG** or **1029** and a *numextents* parameter of **1**. The following **BUILD** command creates such a file, which is then used by the **PREP** command:

```
:BUILD PROGX;CODE=PROG;DISC=,1
```

The USL file created during compilation is a system-defined temporary file **\$OLDPASS**, which is passed directly to the Segmenter. The Segmenter also uses the file **\$OLDPASS**. The prepared program segments are written to it, thus overwriting any existing temporary file of that name. Therefore, if you decide that it is important to save the USL file created during the compilation process, you must create your own permanent program file, referencing it in the **COBOLPREP** command line. It is, however, recommended that you use the default for the *progfile* parameter.

Example

To compile and prepare a COBOL program entered through your current input device with the listing printed on the current list device, enter:

```
:COBOLPREP
```

To compile and prepare a COBOL source program entered from the *textfile* **SFILE** into the program file **MYPROG**, sending the listing to your standard list device, enter:

```
:COBOLPREP SFILE,MYPROG
```

Additional Discussion

COBOL/3000 Reference Manual (32213-90001)

MPE Segmenter Reference Manual (30000-90011)

COBOLII

Compiles a COBOLII program on the COBOL '74 compiler. COBOLII is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
COBOLII [ textfile ] [ , [ uslfile ] [ , [ ;listf ] [ , [ masterfile ] [ , newfile ] ] ] ]  
[ ;INFO=quotedstring ] [ ;WKSP=workspacename ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is COBTEXT. The default is \$STDIN.
<i>uslfile</i>	Actual file designator of the user subprogram library (USL) on which the object program is written. This can be any binary output file with a file code of USL or 1024. Its formal file designator is COBUSL. If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS. If the parameter is entered, it indicates that the file was created in one of four ways: ■ By using the SAVE command to save, the default USL file created during a previous compilation. ■ By building the USL file with the Segmenter command -BUILDUSL. (Refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011).) ■ By creating a new USL file with the BUILD command and specifying a file code of USL or 1024. ■ By specifying a nonexistent <i>uslfile</i> parameter, thereby creating a permanent file of the correct size and type.
<i>listf(ile)</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is COBLIST. The default is \$STDLIST.
<i>masterfile</i>	Actual file designator of the master file with which <i>textfile</i> is merged to produce a composite source. This can be any ASCII input file. The formal designator is COBMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the merged <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. The formal file designator is COBNEW. The default is that no file is written.

quotedstring A sequence of ASCII characters bounded by a pair of single quotation marks or by double quotation marks. You may use the delimiting character as part of the string so long as it appears twice. Any occurrence of two single quotes in a row or two double quotes in a row, is considered part of the string, and not the terminating delimiter.

INFO=*quotedstring* is used in the COBOLII programming language to pass compiler options to a program. These options appear before the first line of source code in the text file. For detailed information, refer to the **RUN** command in this chapter.

workspace-name Actual file designator of an HPToolset workspace used with HPToolset. The formal designator is **COBWKSP**.

Note



The formal file designators (**COBTEXT**, **COBUSL**, **COBLIST**, **COBMAST**, **COBNEW**, and **COBWKSP**) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the **FILE** command.

Use

You may issue this command from a session or a job, but not in **BREAK** or from a program. Pressing **(BREAK)** suspends execution.

Operation

The **COBOLII** command compiles a COBOLII program into a USL file on disk. If you do not specify *textfile*, COBOLII expects the source text to be entered from your standard input device. If you do not specify a *listfile*, COBOLII sends the program listing to the current list device.

Example

To compile a COBOLII program stored in the file **SOURCE** into an object program on the USL file **OBJECT** and send the listing to the disk file **LISTFL**, enter:

```
:BUILD OBJECT;CODE=USL  
:COBOLII SOURCE,OBJECT,LISTFL
```

Additional Discussion

COBOL/II 3000 Reference Manual (32233-90001)

MPE Segmenter Reference Manual (30000-90011)

COBOLIIGO

Compiles, prepares, and executes a COBOLII program on the COBOL '74 compiler. COBOLII is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
COBOLIIGO [textfile] [, [listfile] [, [masterfile] [, newfile] ] ] [ ;INFO=quotedstring ]  
[ ;WKSP=workspacename ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is COBTEXT. The default is \$STDIN.
<i>listfile</i>	Actual file designator of the file on which the program listing is written. This can be any ASCII output file. The formal file designator is COBLIST. The default is \$STDLIST.
<i>masterfile</i>	Actual file designator of the master file which is merged against <i>textfile</i> to produce a composite source. This can be any ASCII input file. The formal file designator is COBMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the merged <i>textfile</i> and the <i>masterfile</i> . This can be any ASCII output file. The formal file designator is COBNEW. The default is that no file is written.
<i>quotedstring</i>	A sequence of ASCII characters bounded by a pair of single quotation marks or by double quotation marks. You may use the delimiting character as part of the string so long as it appears twice. Any occurrence of two single quotes in a row, or two double quotes in a row, is considered part of the string, and not the terminating delimiter. INFO=<i>quotedstring</i> is used in the COBOLII programming language to pass compiler options to a program. These options appear before the first line of source code in the text file. For detailed information, refer to the RUN command in this chapter.
<i>workspace-name</i>	This parameter is the actual file designator of an HPToolset workspace. The formal file designator created by the compiler is COBWKSP.

Note



The formal file designators (COBTEXT, COBLIST, COBMAST, COBNEW, and COBWKSP) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion of “Implicit FILE Commands for Subsystems” under the FILE command.

Use

You may issue this command from a session or job, but not in **BREAK** or from a program. Pressing **(BREAK)** suspends execution.

Operation

The COBOLIIGO command compiles, prepares, and executes a program using the COBOL '74 compiler. If you do not specify *textfile*, COBOLII expects the source program to be entered from your standard input device. If you do not specify *listfile*, COBOLII sends the output to your standard list device.

The USL file created during compilation is a system-defined temporary file, \$OLDPASS, which is passed directly to the Segmenter. The Segmenter purges the USL file and writes the prepared program to \$OLDPASS if the *progfile* is omitted. This file is then executed and may be executed repeatedly.

Example

To compile, prepare, and execute a COBOLII program entered from your standard input device and send the program listing to your standard list device, enter:

```
:COBOLIIGO
```

To compile, prepare and execute a COBOLII program from the disk file TEXTFL and send the program listing to the disk file LISTFL, enter:

```
:COBOLIIGO TEXTFL,LISTFL
```

Additional Discussion

COBOL/II Reference Manual (32233-90001)

MPE Segmenter Reference Manual (30000-90011)

COBOLIIPREP

Compiles and prepares a COBOLII program on the COBOL '74 compiler. COBOLII is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
COBOLIIPREP [ textfile ] [ , [ progfile ] [ , [ listf ] [ , [ masterfile ] [ , newfile ] ] ] ]  
[ ;WKSP=workspacename ] [ ;INFO=quotedstring ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is COBTEXT. The default is \$STDIN.
<i>progfile</i>	Actual file designator of the program file to which the prepared program segments are written. You get the best results by omitting the <i>progfile</i> parameter. When you omit <i>progfile</i>, the Segmenter creates the program file, which reside in the temporary file domain as \$OLDPASS. Its formal file designator is COBPROG. If entered, this <i>progfile</i> indicates that a file was created in one of two ways: ■ By using the BUILD command and specifying a file code of 1029 or PROG, and a <i>numextents</i> value of 1. This file is then used by the PREP command. ■ By specifying a nonexistent file in the <i>progfile</i> parameter. A temporary job file of the correct size and type is created.
<i>listf(ile)</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is COBLIST. The default is \$STDLIST.
<i>masterfile</i>	Actual file designator of the file which is merged against <i>textfile</i> to produce a composite source. This can be any ASCII input file. The formal file designator is COBMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDLIST if <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the file created by merging <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. The formal file designator is COBNEW. The default is that no file is written.
<i>quotedstring</i>	A sequence of ASCII characters bounded by a pair of single quotation marks or by double quotation marks. You may use the delimiting character as part of the string so long as it appears twice. Any occurrence of two single quotes in a row, or two double quotes in a row, is considered part of the string, and not the terminating delimiter. INFO=<i>quotedstring</i> is used in the COBOLII programming language to pass compiler options to a program. These options appear before the first line of source code in the text file. For detailed information, refer to the RUN command in this chapter.

workspace-name This parameter is the actual file designator of an HPToolset workspace used with HPToolset. The formal file designator created by the compiler is COBWKSP.

Note

The formal file designators (COBTEXT, COBLIST, COBMAST, COBNEW, and COBWKSP) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the FILE command.

Use

You may issue this command from a session or a job, but not in **BREAK** or from a program. Pressing **BREAK** suspends execution.

Operation

The COBOLIIPREP command compiles and prepares a COBOLII program into a program file on disk. If you do not specify *textfile*, COBOLII expects your input from your standard input device. If you do not specify *listfile*, COBOLII sends the listing output to your current list device.

The USL file created during compilation is a system-defined temporary file, \$OLDPASS, which is passed directly to the Segmenter. The Segmenter overwrites the USL file and writes the prepared program to \$OLDPASS if *progfile* is omitted. This file can then be executed.

Example

To compile and prepare a COBOLII program entered from your standard input device with the listing printed on the standard list device, enter:

```
:COBOLIIPREP
```

To compile and prepare a COBOLII source program input from the *textfile* SFILE into a program file named MYPROG, with the resulting listing sent to the current list device, enter:

```
:COBOLIIPREP SFILE,MYPROG
```

Additional Discussion

COBOL/II Reference Manual (32233-90001)

MPE Segmenter Reference Manual (30000-90011)

COMMENT

Inserts a comment into the command stream.

Syntax

```
COMMENT [ text ]
```

Parameters

text Information composing the comment text. If the last nonblank character is an ampersand (&), comment text is continued onto the next line. The default is that a record containing only the string "COMMENT" is inserted in the command stream.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **COMMENT** command allows you to include comments or notes in job listings produced on hard copy devices to create headings or explain the purpose of commands or logic used. After **COMMENT** is entered, it can be followed by a message made up of any ASCII characters.

Example

The following example of a job heading employs a comment:

```
!JOB MAC.TECHPUBS
!COMMENT -- THIS IS A SAMPLE JOB
!FORTGO MYPROG
!EOJ
```

CONSOLE

Changes the system console from its current device to another job-accepting terminal.

Syntax

CONSOLE [*ldev*]



Parameters

ldev The logical device number of the new console terminal. If omitted, the CONSOLE command displays the current logical device number of the console.

Use

You may issue this command from a session, job, program, or in BREAK. You may use it to determine the location of the console. To change the location of the console, you must issue the command from the console itself, *unless* it is distributed to users with the ALLOW command, or if you have SM capability.

Operation

The CONSOLE command has two purposes: to display the logical device number of the terminal currently being used as the system console, and to move the location of the console to another logical device. Listing the location of the console requires no special capabilities. To move the console, you require SM capability, unless you are issuing the command from the console or your session has been allowed the use of the CONSOLE command.

The console cannot be moved to a remote terminal, that is, one that is connected to your system via a distributed systems (DS) communications line. Similarly, it should not be moved to a terminal using a multipoint terminal software (MTS) line, or a PAD terminal over a modem.

When you switch the location of the console with the CONSOLE command, a message is printed on the former console and on the logical device receiving the console to display the new logical device number of the system console. The old console is now just another session device and all the console capabilities are transferred to the newly designated terminal.

Without parameters, the CONSOLE command reports the current logical device number of the console. No capabilities are required to execute the command this way.

Note



Before transferring the system console to another terminal, be sure that you can take the console back when desired (ALLOW *user.account;commands=CONSOLE*). Users who are assigned system manager (SM) capability can retrieve the console without being allowed the use of the CONSOLE command.

CONSOLE

Since the system console is a session device, you must be logged on to the console and in a session in order to execute operator commands.

Control and maintenance processor (CMP) and diagnostic control unit (DCU) prompts and messages remain with the configured terminal; for example, channel 1, device 0. This feature cannot be moved to another terminal.

Example

To determine the current location of the system console, enter:

```
:CONSOLE  
CONSOLE IS CURRENTLY ASSIGNED TO LDEV 20
```

To transfer the console to the terminal identified by MPE V/E as logical device #31, enter:

```
:CONSOLE 31  
CONSOLE HAS BEEN SWITCHED FROM LDEV 20 TO LDEV 31
```

CONTINUE

Overrides job errors so that the job continues executing.

Syntax

```
CONTINUE
```

Parameters

None.

Use

You may issue this command in a session, job, or in a **BREAK**, but not from a program. Pressing **BREAK** has no effect on this command.

Operation

The **CONTINUE** command permits a job or session to continue even though the command immediately following **CONTINUE** results in an error (with accompanying error message). It is not needed in a session because sessions do not terminate when a command error occurs. The **CONTINUE** command is typically used in the line preceding any command suspected of causing the job to abort. If an error does occur, the job continues to run, and the job output contains the error message.

Example

If you anticipate a possible error resulting from the command **RUN MYPROG**, and you wish to override this error and allow the job to continue executing, enter:

```
!JOB MAC.PUBS
!CONTINUE
!RUN MYPROG
!RUN MYPROG2
!EOJ
```

Additional Discussion

Refer to chapter 3 of this manual.

DATA

Enters data into the system from any device file except \$STDIN.

Syntax

```
DATA [jsname,] username[/userpass].acctname [/acctpass] [;filename]
```

Parameters

<i>jsname</i>	Name of job or session that is to read data. The default is no job or session name.
<i>username</i>	User name that allows you to access MPE V/E under this account, as established by the account manager.
<i>userpass</i>	User password, optionally assigned by the account manager.
<i>acctname</i>	Account name under which the job or session is running, as established by the system manager.
<i>acctp</i> ass	Account password, optionally assigned by the system manager.
<i>filename</i>	Additional qualifying name for the data that can be used by a job or session to access data. It may be used to distinguish two separate data decks read from different card readers by the same program. The default is that no distinguishing name is assigned.

Note



The *jsname*, *username*, *userpass*, *acctname*, *acctp*ass, and *filename* parameters all contain up to eight alphanumeric characters, beginning with an alphabetic character.

Use

You may issue this command from a session or job, but not from \$STDIN. You may not use it from a program or in BREAK. Pressing **BREAK** has no effect on this command.

Operation

This command identifies input that is to be read from a device file other than the standard input device. You would use the DATA command to read a file from a spooled input device while running an interactive session. However, if you enter the DATA command from the terminal during a session, the terminal becomes a data file for some unspecified process, which prevents you from using it until a process issues a request for data from the terminal.

The DATA command is the only command that you can enter before a job or session is initiated with the JOB or HELLO commands. Files identified by data may be input on cards, magnetic tape, paper tape, or from a terminal. To designate a set of data as an auxiliary file for a job or session, enter the DATA command followed by the data deck or file, and the :EOD command.

Begin the job or session using the same identity (*[jsname,] username.acctname*) used in the **DATA** command so that the job or session can use the data. If the *filename* parameter is omitted, several data decks or files can be read from any job or session with the same identity.

You must use the **STREAM** command to submit data from a disk file. When data is entered from magnetic tape, the file must reside on a single tape volume with a blocking factor of 1. When the medium containing the data file is placed on the input device and that device is placed online, MPE V/E reads the entire file if the input device is spooled. Only the **DATA** command is read if the file is input from a nonspooled device. At that point, the job can access the data, which remains available until it is actually read.

Example

A session is running and is identified by the session name **SESSA**, user name **BROWN**, and account name **ACCT1**. You want to make data on punched cards available to that session, to be used by a program named **PROGY**. This program references the data under the formal file designator **DATAFL**. To do so, follows these steps:

1. Arrange the data deck beginning with the **DATA** command and terminating with the **:EOD** command, as follows:

```
DATA SESSA,BROWN.ACCT1
```

```

      .
      .
      .
    data
      .
      .
      .

```

```
:EOD
```

2. Load the cards into the card reader and make the reader ready. If the card reader is a spooled device, the data is copied to disk, where it awaits access. If it is not a spooled device, the **DATA** command is read, but the subsequent data remains in the card reader until the program accesses it. The **DATA** command is used to build an entry in the device directory that identifies the file to the system.
3. Begin your session, making sure you use exactly the same session identity you entered in the **DATA** command. (For instance, if you omit if the optional session name, the session cannot access the data.) To logon, enter:

```
:HELLO SESSA,BROWN.ACCT1
```

4. Enter a **FILE** command equating the formal file designator **DATAFL** (used by the program **PROGY**) with the card reader. For example:

```
:FILE DATAFL;DEV=CARD
```


DATA

5. Run the program that requires the data; when the program attempts to open the file DATAFL, it is available:

:RUN PROGY

If the data has not been entered through the card reader before this step, the program transmits a message to the system operator requesting this data and waits for it to become available. The system operator must then decide whether to supply the data via the DATA command or, if the data is not available, to send a message asking you to supply it, or abort the session.

6. Once the data has been read, it is no longer available to the system. If another program requires it, the data must be entered again.

In the next example, a data file is created on disk and the STREAM command is used to make the file available to your program:

1. Create the file DATAFL on disk beginning with the DATA command and ending with the :EOD command.

DATA SESSB,BROWN.ACCT1

.
.
.
data

.
.
.
:EOD

2. Stream the data file, using the STREAM command as follows:

:STREAM DATAFL

3. Log onto a session, making sure that you use exactly the same session identity as you entered in the DATA command:

:HELLO SESSB,BROWN.ACCT1

4. Enter a FILE command equating the formal file designator (used by the program) with the streams device (identified by the device class name JOBTape):

:FILE DATAFL;DEV=JOBTape

5. Run the program that requires the data:

:RUN PROGY

DEALLOCATE

Deallocates a program or procedure previously loaded into memory with the **ALLOCATE** command.

Syntax

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

Parameters

PROGRAM	The program file indicated by <i>name</i> is deallocated. This is the default.
PROCEDURE	The code segment containing the procedure specified by <i>name</i> in SL.PUB.SYS is deallocated. The default is PROGRAM .
<i>name</i>	The name of the program file or procedure to be deallocated.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. Any autoallocated program for which you have **EXECUTE** access can be deallocated. If you have **OP** capability, you can deallocate any program. However, you must have **OP** capability to deallocate any programs that have been permanently allocated via the **ALLOCATE** command.

Operation

The **DEALLOCATE** command immediately releases table entries belonging to a program file or procedure that has been allocated. If the program is currently executing, the command takes effect once the program or procedure is no longer in use.

Example

To deallocate a program with the file name **PROGEX**, enter:

```
:DEALLOCATE PROGEX
```

The **DEALLOCATE** command does not give back memory, it gives back table entries.

DEBUG

Invokes the MPE V/E debug facility.

Syntax

DEBUG

Parameters

None.

Use

You may issue this command from a session, program, or in **BREAK**. You may not use it from a job. Pressing **(BREAK)** has no effect on this command. You must have privilege mode (PM) capability to use this command.

Operation

The **DEBUG** command, which you can only execute with privileged mode (PM) capability, is an extension of the MPE V/E debug facility. System programmers use it primarily to analyze code and trace errors in program logic. **DEBUG** is fully explained in the *MPE Debug/Stack Dump Reference Manual*(30000-90012).

Caution



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

DELETESPOOLFILE

Deletes a spool file from disk.

Syntax

$$\text{DELETESPOOLFILE} \left\{ \begin{array}{l} \#0nnn \\ \#Innn \\ ldev \end{array} \right\}$$

Parameters

- #0nnn** The identification of a ready or active output spool file.
- #Innn** The identification of a ready, spooled data input file.
- ldev** The logical device number on which the spool file is active.

Use

You may issue this command from a session, program, or in **BREAK**. You may not use it from a job. Pressing **BREAK** has no effect on this command. You may use this command only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

Before deleting an active spool file, you should first take the output device offline. This allows you time to enter the command and determine that the active spool file corresponds to the correct output device. When MPE V/E returns the colon prompt (:), you know that the **DELETESPOOLFILE** instruction has been sent to the spooler process. It is not executed, however, until the output device is put back online.

You may not use the **DELETESPOOLFILE** command on the following files:

- System-defined standard input spool files (**\$STDIN**). Delete them with the **ABORTJOB** command.
- Active spool files with data input, entered via the **STREAM** command. You may delete these only when they are ready. You may not delete these files when they are open.

The **DELETESPOOLFILE** command deletes active data input files which are submitted on a spooled device, such as a card reader. It cannot delete such files while they are being streamed.

DELETESPOOLFILE

Example

To delete the active spool file being printed on logical device #6, first take the printer offline. This generates a **NOT READY** message at the console, after which you may enter the **DELETESPOOLFILE** command, as shown below:

```
11:21/7/LDEV#6 NOT READY  
:DELETESPOOLFILE 6
```

When you put the device back online, the trailer page is printed and the file deleted. If you have suppressed header/trailer output with the **HEADOFF** command, no trailer is printed before the spool file is deleted; however, the printer skips to the top of the next physical page. If the device is a page printer, the default environment is reloaded.

DISALLOW

Prohibits access to a specific operator command.

Syntax

```
DISALLOW FILE=formaldesignator[ ;SHOW ] DISALLOW { @.@
user.@
@.acct
user.acct }
;COMMANDS=command[ ,command, ... ]
```

Parameters

<i>formaldesignator</i>	A formal ASCII filename, which may consist of up to eight alphanumeric characters, beginning with an alphabetic character.
SHOW	Lists input lines on \$STDLIST.
@.@	Prohibits access to all users, whether logged on or not.
<i>user</i> .@	Prohibits access to a specific user in all accounts.
@. <i>acct</i>	Prohibits access to all users in a specific account.
<i>user</i> . <i>acct</i>	Prohibits access to a specific user in a specific account.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may use this command only from the console *unless* it is distributed to users with the **ALLOW** command.

Operation

The system operator uses the **DISALLOW** command to prevent a user from executing specific console commands previously allowed with the **ALLOW** command. One form of the command requires you to enter the list of prohibited commands directly at the console. The second form of **DISALLOW** is the indirect mode, in which you use **EDIT/3000** to create a file that contains the user name and account of those users who are prohibited from executing certain operator commands, and a list of disallowed commands.

DISALLOW

The procedure is illustrated in the following example:

```
:EDITOR
HP32201A.7.16 EDIT/3000  TUE, JUL 11, 1989,  5:08 PM
(C) HEWLETT-PACKARD CO. 1984
/ADD
  1 SUSAN.PAYROLL;COMMANDS=ALTJOB,ALTSPoolFILE
  2 JOHN.ACCTNG;COMMANDS=ALTSPoolFILE,DELETESPoolFILE
  3 //
...
/KEEP COMNDTMP
/E
```

Once this file is defined, command input is taken from it rather than from data entered at the console. No continued records are permitted. You can then execute the **DISALLOW** command by entering the following:

```
:DISALLOW ENTITY=COMNDTMP;SHOW
```

The ;SHOW parameter directs MPE V/E to display each command line as it is executed from the file.

A third form of the command is the subsystem mode. In this case, you need only type the command name, **DISALLOW**, followed by Return. You are then prompted with a >, after which you enter command parameters. Subsystem mode ends when an EOF is received, or when you exit.

Example

To prohibit the user **MAC.TECH** from executing the **REPLY** and **ABORTIO** commands, enter the following on the system console:

```
:DISALLOW MAC.TECH;COMMANDS=REPLY,ABORTIO
```

To use subsystem mode to prohibit the user **MGR.MANUALS** from executing the **BREAKJOB** command, enter the following on the system console:

```
:DISALLOW
>MGR.MANUALS;COMMANDS=BREAKJOB
>EXIT
:
```

DISASSOCIATE

Removes control of a device from the user.

Syntax

`DISASSOCIATE devclass`

Parameter

devclass The name of a device class configured during SYSDUMP.

Use

You may issue this command from a session, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **DISASSOCIATE** command negates a previously issued **ASSOCIATE** command by removing control of a device class from a user. The system operator or the user may issue the command. The user implicitly disassociates a device when logging off.

Example

To terminate control of the device class **TAPE**, enter:

```
:DISASSOCIATE TAPE
```

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005).

DISCRPS

Enables or disables the rotational position sensing (RPS) feature on a specified logical device. It requires a special firmware upgrade CS-80 disk drive.

Syntax

$$\text{DISCRPS } ldev \left\{ \begin{array}{l} \text{,ENABLE} \\ \text{,DISABLE} \end{array} \right\}$$

Parameters

ldev The logical device number of the specified CS-80 disk drive.

ENABLE Enables rotational position sensing on the device.

DISABLE Disables rotational position sensing on the device.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You may execute it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The **DISCRPS** command allows you to enable or disable the rotational position sensing feature for CS-80 disk drives. With RPS enabled, the disk drive signals its availability to do an I/O only when it is a small rotational distance away from the target data. This improves system performance when more than one drive is connected to the same HP-IB channel.

Example

To enable the RPS feature on logical device #1 and display the drive's status, enter:

```
:DISCRPS 1,ENABLE
:SHOWDEV 1
LDEV      AVAIL      OWNERSHIP    VOID    DEN    ASSOCIATION
1         DISC   (RPS) 50 FILES
```

DISMOUNT

Causes a volume set that was explicitly mounted by the user to be dismounted.

Syntax

$$\text{DISMOUNT} \left[\left\{ \begin{array}{c} * \\ vcsname \end{array} \right\} \left[.groupname \left[.acctname \right] \right] \right]$$

Parameters

- * or vcsname* Specifies the volume set. The asterisk (*) specifies the home volume set for the group and account specified, or the logon group and account if they are not specified. The *vcsname* parameter specifies the volume set or class name of a previously defined volume set or volume class. Omitting this parameter is equivalent to using ***.
- groupname.*
acctname Specifies the group and account which created the volume set. The default is the logon group and account.

Use

You may issue this command from a session, job, or in **BREAK**. You may not use it from a program. Pressing **BREAK** has no effect on this command.

Operation

The **DISMOUNT** command allows you to dismount a volume set that you explicitly mounted using the **MOUNT** command. If there are no other users of the volume set, the devices on which it resides are made available to the system. You can request a dismount only for a volume set that you have mounted; you cannot alter the status of the volume set for other users.

Example

To dismount the volume set **MYSET** that you previously mounted via a **MOUNT** command, enter:

```
:DISMOUNT MYSET
```

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005)

DOWN

Removes a device from normal system use.

Syntax

`DOWN ldev`

Parameter

ldev The logical device number of the device being taken offline.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may use it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

When you issue the **DOWN** command for a device that is in use, the request is satisfied when the process currently accessing it releases it.

Once you down a device corresponding to the master volume of a non-system domain disk drive, MPE V/E rejects all further requests to use any volume in the set to which the master volume belongs.

The system console cannot be downed. Any attempt to do so results in the following error message:

`DOWN NOT PERFORMED ON CONSOLE DEVICE (CIERR 3150)`

Caution



When any device is powered down without the use of the **DOWN** command, subsequent access to that device can result in indefinite waiting, erroneous transfers, or other incorrect operation. Often these failures occur with no indication to the system operator or to the user. For this reason, it is very important that every device which is not fully operational (especially those which are powered down) be downed with the **DOWN** command. A device that is inoperable for more than a few hours can be temporarily removed from the I/O configuration at system startup.

Example

To take logical device #7 offline, enter:

:DOWN 7

To take logical device #10 (an input-spooled, job-accepting magnetic tape) offline, enter:

:DOWN 10

:STOPSPool 10

11:16/31/SP#10/STOPPED

11:16/31/LDEV#10 NOT READY



DOWNLOAD

Downloads format information to a line printer.

Syntax

```
DOWNLOAD ldev [ , filename ] [ , ... ]  
                [ , MARGIN=nn ]
```

Parameters

<i>ldev</i>	The logical device number of the output device. This device must be a HP 2608 or HP 2563 line printer.
<i>filename</i>	The fully qualified name of a file containing the download control information.
<i>nn</i>	The print position that the first byte of data assumes. This number can be between 1 and 16 inclusive.

Note



The HP 2608 hardware documentation discusses a margin offset which varies from 0 to 15. This offset is not relevant to the margin parameter of the **DOWNLOAD** command, as the software compensates for the hardware offset of *nn* -1.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may issue it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The system operator uses the **DOWNLOAD** command to transmit format information to system printers only. It cannot be used with remote printers.

The vertical format control (VFC) image file (*filename*) can define the margin setting as well as the VFC image on an HP 2608A or HP 2608S line printer. The number of print lines per form is limited to 127. Although the HP 2608S printer recognizes the **DOWNLOAD** command, Hewlett-Packard recommends that you control the HP 2608S with an environment file rather than the **DOWNLOAD** command. You cannot download a VFC file to HP 2631B printer; only the **MARGIN=*nn*** parameter is allowed. For a discussion of special hardware considerations, refer to section VIII of the *MPE System Operation and Resource Management Manual* (32033-90005).

If the **MARGIN=*nn*** parameter is specified on an HP 2608A or HP 2608S, and a **MARGIN** record has also been specified in the VFC file, the **MARGIN** record in the VFC file overrides the **MARGIN** parameter of the **DOWNLOAD** command. This parameter should only be used in cases where there is no **MARGIN** record in the VFC file.

When a particular print job has requirements for forms and/or a VFC file, you indicate this need via a **FORMS** message (refer to the examples below).

Caution

Do not issue a **DOWNLOAD** command to an HP 2608S while a spool file is active. This will make the device unavailable, and it will remain so until the system is restarted with a **WARMSTART**.

Example

To respond to a forms message, enter:

```
IO/15:46/22/FORMS: PLEASE MOUNT PAYCHECK FORMS. USE VFC=VFCPAY  
IO/15:46/22/SP#11/LDEV# FOR #S93;OUTFILE ON HP 2608 (1)
```

```
:DOWNLOAD 11,VFCPAY
```

To reset the VFC to its original state, you must reference a file that contains default specifications (such as **VFC6** in the following example) by entering:

```
:DOWNLOAD 11,VFC6.PUB.SYS
```

To set the left margin print position to column 4 (the installation-defined default), enter:

```
:DOWNLOAD 11,MARGIN=4
```

DSTAT

Displays the current status of the disk drives on the system.

Syntax

```
DSTAT  $\left[ \begin{array}{l} ldev \\ ALL \end{array} \right]$ 
```

Parameters

- ldev* An integer specifying the logical device number of the disk drive whose status you want to know.
- ALL** Displays the status of all disk drives, both system and non-system. The default is that only the status of non-system disks is displayed.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **DSTAT** command is used to display the current status of one or more disk drives on the system. For example:

:DSTAT ALL

LDEV-TYPE	STATUS	VOLUME (VOLUME SET-GEN)
-----	-----	-----
1-7933/7935	SYSTEM	MARKET1
2-7920	SYSTEM	MARKET2
3-7905(R)	AVAIL	(USERVOL1.PUB.SYS-0)
4-7920	DOWNED	
5-7920	AVAIL	(SLAVOL1.PUB.SYS-0)
11-7920	AVAIL	(SLAVOL2.PUB.SYS-0)
12-7920	AVAIL	(SLAVOL3.PUB.SYS-0)

The status can be:

AVAIL	A non-system disk is physically mounted, but is not being used.
AVAIL *SCRATCH*	A non-system disk has one bit modified in the disk label, so that it cannot be used. You can use the VINIT subsystem to reset this bit. The disk then assumes its former state (either serial disk or private volume).
DOWNED	The disk is not available for use.
FOREIGN	The disk is unformatted or contains data in a form not recognizable by MPE V/E.
MOUNTED	The disk is mounted and is currently being accessed.
OFF LINE	The disk is spun down.
SYSTEM	The disk is allocated to the operating system.
SERIAL	This is a serial disk.
DOWN-PND	A DOWN command has been issued for this disk to make it unavailable for use. However, since the disk was still in use when the command was issued, the DOWN is pending completion.
RESERVED	A non-system disk has been reserved for use by a process.

Example

To display the status of LDEV 7 enter:

```
:DSTAT 7
```

LDEV-TYPE	STATUS	VOLUME (VOLUME SET-GEN)
-----	-----	-----
7-7902	FOREIGN	*UNALLOCATED*

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005).

EDITOR

Runs EDIT/3000, which is used to create and manipulate ASCII files.

Syntax

EDITOR [*listfile*]

Parameters

listfile Actual file designator of file to receive any output resulting from the EDIT/3000 commands LIST and XPLAIN when the OFFLINE option is specified. It can be any ASCII output file. The formal file designator and default is EDTLIST. If specified with no device parameter, the default device is LP.

Note



The formal file designator used in this command, EDTLIST, cannot be backreferenced as an actual file designator in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the MPE V/E FILE command in this chapter.

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

Operation

The EDITOR command runs the EDIT/3000 subsystem.

Example

To run EDIT/3000 during a session and specify a line printer (device class LP) as the list device for offline output, enter:

```
:FILE LISTFILE;DEV=LP  
:EDITOR *LISTFILE
```

Because the *listfile* is often a line printer, it is often defined in an MPE V/E FILE command and backreferenced as in the preceding example.

Additional Discussion

EDIT/3000 Reference Manual (03000-90012)

ELSE

Provides an alternate execution sequence for an IF statement.

Syntax

```
ELSE
```

Parameters

None.

Use

You may issue this command from a session, job, or in **BREAK**. You may not use it from a program. Pressing **BREAK** has no effect on this command.

Operation

The **ELSE** command is used only in conjunction with the **IF** command. The **IF** command is used with the **ENDIF** command, and optionally with the **ELSE** command, to control the execution of a job. The **IF**, **ENDIF**, and optional **ELSE** commands constitute an IF block. A logical expression is evaluated, and if true, the IF block is executed; if false, the **ELSE** block (if one exists) is executed.

Example

The following job listing illustrates the use of the **ELSE** command:

```
!CONTINUE
!SPL MYPROG,MYUSL
!IF JCW>=FATAL THEN
!   TELL MAC.TECHPUBS;COMPILE FAILED
!ELSE
!   TELL MAC.TECHPUBS;COMPILE COMPLETED
!ENDIF
```

For more information on job control words (JCWs), refer to **SETJCW** and **SHOWJCW** in this chapter.

ENDIF

Terminates an IF block.

Syntax

```
ENDIF
```

Parameters

None.

Use

You may issue this command from a session, job, or in **BREAK**. You may not use it from a program. Pressing **BREAK** has no effect on this command.

Operation

The **ENDIF** command is used to terminate an IF block. The **IF** command, the optional **ELSE** command, and the **ENDIF** command constitute an IF block. They are used to control the execution of a job. A logical expression is evaluated, and if true, the IF block is executed; if false, the **ELSE** block (if one exists) is executed. The example on the preceding page illustrates how this command may be used.

:EOD

Denotes end-of-data on an input stream and terminates a set of data initialized by the **DATA** command.

Syntax

:EOD

Parameters

None.

Use

You may issue this command from a session or a job. You may not use it from a program or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **:EOD** command is used to signify the end of a data set entered through a standard input device. It is also used to delimit data entered via the **DATA** command on data-accepting devices such as tape drives and card readers.

In most cases, programmers use **:EOD** for delimiting data. Any record beginning with a colon, however, will delimit data. Using a command other than **:EOD** for this purpose depends upon whether the standard input file is opened with the filename **\$STDIN** or **\$STDINX**. Refer to the table on the following page to see which delimiters are permitted with which files. If you enter certain end-of-file indicators, such as **:EOD** or **:EOF**, interactively while using MPE V/E or from a subsystem, you may terminate your session.

When using a compiler language that does not provide a convention for terminating compilation (such as **END** in **SPL**), you must enter **:EOD** after the last record of your source program to ensure proper delimiting of your input. You do not require the **:EOD** command when using the **BASIC** interpreter since the subsystem provides different conventions for delimiting data.

The **:EOD** command causes the **FREAD** intrinsic to return the **CCG** condition code to the calling program, which indicates the end-of-file condition on the terminal.

Example

To terminate a data file entered on cards for a session identified as **SESS1,BLACK.ACCTSP**, your data deck would contain **:EOD** as the last record. Enter:

DATA SESS1,BLACK.ACCTSP

.
.
.
.
.

data

:EOD

:EOD

The following FORTRAN program is an example of how you can use the :EOD command to terminate a data set entered through a standard input device:

FORTRAN

PAGE 0001 HP32102B.01.04 (C) HEWLETT-PACKARD CO. 1985

```
>$CONTROL USLINIT
> PROGRAM MONEY
> INTEGER QUARTERS,DIMES,NICKELS,PENNIES
> DISPLAY "INPUT MONEY AMOUNT IN DECIMAL FORM "
> ACCEPT DECIMALFORM
> CALL CHANGER(DECIMALFORM,QUARTERS,DIMES,NICKELS,PENNIES)
> DISPLAY QUARTERS," QUARTERS"
> DISPLAY DIMES," DIMES"
> DISPLAY NICKELS," NICKELS"
> DISPLAY PENNIES," PENNIES"
> STOP
> END
```

PROGRAM UNIT MONEY COMPILED

```
> SUBROUTINE CHANGER(DECIMALFORM,QUARTERS,DIMES,NICKELS,PENNIES)
> INTEGER QUARTERS,DIMES,NICKELS,PENNIES
> DECIMALFORM = DECIMALFORM*100
> QUARTERS = DECIMALFORM/25
> REMAINDER = DECIMALFORM-(QUARTERS*25)
> DIMES=REMAINDER/10
> REMAINDER=REMAINDER-(DIMES*10)
> NICKELS=REMAINDER/5
> PENNIES=REMAINDER-(NICKELS*5)
> RETURN
> END
```

PROGRAM UNIT CHANGER COMPILED

>:EOD

```
****      GLOBAL STATISTICS      ****
****      NO ERRORS,   NO WARNINGS      ****
TOTAL COMPILE TIME  0:00:01
TOTAL ELAPSED TIME   0:01:29
```

END OF COMPILE

:EOF:

Simulates a physical end-of-file on an input stream from any device. The colons (:) are a required part of this command.

Syntax

:EOF:

Parameters

None.

Use

You may issue this command from a session or job. You may not use it from a program or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **:EOF:** command denotes the end-of-file when read through an input stream. It is required to simulate a physical end-of-file on an HP 2894A Card Reader/Punch when that device is configured to refuse data submitted via a **DATA** command. (This device does not provide a true hardware end-of-file indication.)

Note

If you use this command interactively, it will terminate your session.



Example

To denote the end-of-file for a file read through an input stream, enter:

:EOF:

The last colon in this command must be followed by a blank or **Return**.

Additional Discussion

For more information on end-of-file indicators refer to the **:EOD** command in this chapter.

EOJ

Ends a batch job.

Syntax

EOJ

Parameters

None.

Use

You may issue this command from a job. You may not use it from a session, program or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **EOJ** command terminates a batch job and displays the CPU time (in seconds) and the elapsed time since the beginning of the job (rounded to the nearest minute). MPE V/E also adds the CPU time and file space used by your job to the resource usage counters maintained for your logon account and group.

If you omit the **EOJ** command from a job, the next **JOB** command terminates the current job and starts a new one. The end of the first job is indicated by the standard end-of-job display, and the beginning of the next job is denoted by the normal job initiation display.

Example

The following example shows how **EOJ** is used within a job file to terminate a batch job:

```
!JOB MAC.PUBS
!RUN MYPROG1
!RUN MYPROG2
!EOJ
```

FCOPY

Runs the FCOPY subsystem.

Syntax

FCOPY [*fcopycommand*]



Parameters

fcopycommand An FCOPY/3000 subsystem command. The FCOPY/3000 subsystem enables you to copy files or selected portions of files from any supported input device to any supported output device. There are many subsystem commands; only the most common are used in the “Examples” for this command. Refer to the *FCOPY Command Reference Manual* (32212-90003) for more information.

Use

You may issue this command from a session or a job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The FCOPY command runs the FCOPY subsystem from MPE V/E. If you enter the command with no parameters, FCOPY prompts you via the “>” prompt for subsystem commands until you enter an **EXIT** command. If you use the *fcopycommand* parameter, FCOPY executes the FCOPY subsystem command and then returns control to MPE V/E.

Example

To access the FCOPY subsystem and execute multiple commands, enter:

```
:FCOPY
HP32212A.3.12 FILE COPIER (C) HEWLETT-PACKARD CO. 1985
>
```


FCOPY

To access FCOPY to execute a single command and return control to MPE V/E, enter the command as follows:

```
:FCOPY FROM=UDC.TECHPUBS;TO=TEMP;NEW  
HP32212A.3.19 FILE COPIER © HEWLETT-PACKARD CO., 1984.
```

```
EOF FOUND IN FROMFILE AFTER RECORD 23
```

```
24 RECORDS PROCESSED *** 0 ERRORS
```

```
END OF SUBSYSTEM  
:
```

Additional Discussion

FCOPY Commands Reference Manual (32212-90003)

FILE

Declares the file attributes to be used when a file is opened. You may use this declaration, informally known as a file equation, to override programmatic or system default file specifications. With the addition of shared parameters from NS/3000 AdvanceNet or the DS/3000 subsystems, the declaration may specify a formal file designator that you can use to access a remote file or device in a subsequent command or intrinsic. NS/3000 AdvanceNet and DS/3000 are not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```

FILE formaldesignator
    [
        =*formaldesignator
        =$NULL
        =$NEWPASS
        =$OLDPASS
        =$STDIN
        =$STDINX
        =$STDLIST
        =filereference[ :nodespec ] [ ,filedomain ]
    ]
    [ ;DEV=[ [ envname ] # ] [ device ] [ ,outpri ] [ ,numcopies ] ] [ ;VTERM ]
    [ ;ENV=envfile [ :nodespec ] ] [ ;option ] [ ;access ] [ ;disposition ]

```

Parameters

formaldesignator

A formal file designator that has the form

```
filename[.groupname[.accountname] ] [ :nodespec ]
```

in which *filename*, *groupname*, and *accountname* are the identifiers that form a fully qualified file name. Each identifier may contain from one to eight alphanumeric characters, beginning with an alphabetic character. This file name may be used to identify the file in subsequent commands or intrinsic calls.

The *nodespec* extension of the formal file designator, explained below, is a parameter shared with the NS/3000 AdvanceNet subsystem and the DS/3000 subsystem. It is not part of the fundamental operating system (see the “Note” below). MPE V/E accepts this extended formal file designator with a node specification following a colon (:) in **FILE** and **RESET** and in the **FOPEN** intrinsic. Refer to the **RESET** command in this chapter and also the *MPE V Intrinsics Reference Manual* (32033-90007).

You should be aware that the **FOPEN** intrinsic recognizes only *filename* in the *formaldesignator*. **FOPEN** ignores any *groupname* or *accountname* specifications.

FILE

If *formaldesignator* is not equated to another file designation, the parameter specifies the name of an actual file. Placing an asterisk ahead of the parameter (**formaldesignator*) establishes a backreference to a formal file designator defined in a previously issued FILE command.

The default is a temporary, nameless file that may be written to but not saved.

\$NULL Actual file designator of a system-defined file that is always treated as an empty file. When **\$NULL** is accessed by a program for input, that program receives only an end-of-file indication. When it is accessed by a program for output, the associated **WRITE** request is accepted by MPE V/E, but no physical output is actually performed.

\$NEWPASS A system-defined temporary job file. When **\$NEWPASS** is closed, it is referenced by the name **\$OLDPASS**. Opening **\$NEWPASS** destroys any previous temporary **\$NEWPASS** file.

\$OLDPASS The system-defined name of the last temporary file that was closed as **\$NEWPASS**.

\$STDIN The system-defined name of the standard job input device. A colon (:) as the first character read on this file indicates end-of-file.

\$STDINX The same as **\$STDIN** except that a colon can be read as the first character and received as data. :EOF: and :EOD statements cannot be read as data.

\$STDLIST The system-defined name for the standard job or session limit device.

filereference The actual file designator of the file, which has the form

filename[/lockword][.groupname[.accountname]]

Each identifier may contain from one to eight alphanumeric characters, beginning with an alphabetic character. In a batch job, the file will fail to open if the file has a *lockword* that is not specified in *filereference*. In a session, MPE V/E will prompt you for a *lockword* if one exists.

nodespec An extension of the formal file reference. It may be an environment identification (specified in a previous **DSL** or **REMOTE** command), or it may be **\$BACK**. It may appear in the formal file designator of the file or as an extension of an actual file reference. If an environment identification appears in a file designation and in the **DEV=** option, an attempt to open the file (with the **FOPEN** intrinsic, for example) produces an error.

\$BACK instructs MPE V/E to “hop backward” one node toward your local system to find the specified file. This works only if the **FILE** command is issued in a remote session. If the systems involved are connected in a local area network (LAN), one “hop backward” always means returning to your local system. The **\$BACK** specification is the same as **DEV=#** without an environment name.

Note

The *nodespec* parameter and the *DSL*INE and *REMOTE* commands are not part of the HP 3000 Fundamental Operating System. The NS/3000 AdvanceNet subsystem or the DS/3000 subsystem must be purchased separately. The *nodespec* parameter is optional: if you do not have NS/3000 AdvanceNet or DS/3000, omitting the *nodespec* parameter will make no difference in the performance of the *FILE* command.

If you specify *nodespec* on a system that does not have NS/3000 or DS/3000, you get an error. The *nodespec* parameter is controlled by the NS/3000 or DS/3000 subsystem. Refer to the *NS/3000 User/Programmer Reference Manual* (32344-90001) or the *DSN/DS User/Programmer Reference Manual* (32189-90001).

filedomain

The domain of the file, which may be *OLD*, *NEW*, or *OLDTEMP*.

- | | |
|----------------|--|
| NEW | Creates a new file, which becomes the default. The <i>NEW</i> file may be permanent or temporary, depending on how the file was created. You must use either the <i>BUILD</i> command or the <i>FOPEN</i> intrinsic to create the file. Refer to the <i>BUILD</i> command in this chapter. |
| OLD | Specifies an existing permanent file that was saved in the system or in the private volume domain. The file continues to exist after the current job or session ends. |
| OLDTEMP | Specifies a temporary file that already exists in the temporary session or job file domain. The file is deleted at the end of the current job or session. |

envname

This may be a *nodespec*, a DS device class name, logical device number, or an X.25 node name. Over the DS-compatible links, the logical device number may not be preceded by the *DSLDEV* prefix. Instead, use just the logical device number. The parameter *envname* may consist of up to eight alphanumeric characters, beginning with an alphabetic character.

Note

The *envname* parameter is not part of the HP 3000 Fundamental Operating System. The NS/3000 AdvanceNet subsystem or the DS/3000 subsystem must be purchased separately. The *envname* parameter is optional: if you do not have NS/3000 AdvanceNet or DS/3000, omitting the *envname* parameter will make no difference in the performance of the *FILE* command.

If you specify *envname* on a system that does not have NS/3000 or DS/3000, you get an error. The *envname* parameter is controlled by the NS/3000 or DS/3000 subsystem. Refer to the *NS/3000 User/Programmer Reference Manual* (32189-90001) or the *DSN/DS User/Programmer Reference Manual* (32344-90001).

FILE

<i>device</i>	The logical device name or logical device number of a device, such as a disk, tape, printer, or a terminal. The default is DISC. If the DEV= option appears, it must be followed by at least one parameter (the parameter can simply be #). If you are opening a file that is to reside on a private volume, you must specify a device class that includes the drives upon which the home volume set is mounted. The file is then allocated to any of the volume set's volumes that fall within that class. The default device class is DISC. A previously defined environment identifier is permitted in the DEV= option, but the <i>domain</i> and <i>organization</i> qualifiers are not permitted.
<i>outpri</i>	The output priority requested for a spooled device. This may have a value of 1 (the lowest priority) to 13 (the highest).
<i>numcopies</i>	The number of copies requested for a spooled output devicefile. The maximum number is 127.
VTERM	Instructs MPE V/E to use the reverse virtual terminal service instead of the remote file access. You may use VTERM only if the designated device is a remote terminal. Using VTERM allows a local application to perform I/O to remote terminals located on systems which support reverse virtual terminals.
<i>envfile</i>	The name of a file containing laser printer environment information, which controls the printing output formats on the laser printer. This name may be an actual file designator, or it may be a formal file designator preceded by an asterisk (*). The information in this file contains such specifications as page size, the character font, forms, and other printer requirements to be used with the HP laser printing system, entered with IDS/3000 and IFS/3000. The file must be in a form suitable for downloading to the HP 2680A, 2688, 2563A, or 2608S. To specify an environment file called TRADMIN.HPENV.HP2680A for a form printed on the HP2680A, enter: <u>:FILE LP;DEV=LASER;ENV=TRADMIN.HPENV.HP2680A</u> Refer to the <i>IFS/3000 Reference Guide</i> (36580-90001) for information on creating environment files. The ENV= parameter in the FILE command overrides the environment specified in the FOPEN intrinsic. If you use the ENV= parameter and omit the <i>*formalfiledesignator</i> or <i>filereference</i>, any environment file specification in a subsequent FOPEN of the device file is ignored.
<i>option</i>	Any valid option for the FILE command.

Syntax for *option*

$$\left[\begin{array}{l} \text{;REC}=[\text{recsize}] \left[\begin{array}{l} ,[\text{blockfactor}] \left[\begin{array}{l} ,\left\{ \begin{array}{l} \text{F} \\ \text{U} \\ \text{V} \end{array} \right\} \left[\begin{array}{l} \text{BINARY} \\ \text{ASCII} \end{array} \right] \end{array} \right] \end{array} \right] \end{array} \right] \left[\begin{array}{l} \text{;DEN}=[\text{density}] \\ \text{;DISC}=[\text{numrec}] \left[\begin{array}{l} ,[\text{numextents}] \left[\begin{array}{l} ,[\text{initialloc}] \left[\begin{array}{l} \text{;CODE}=[\text{filecode}] \left[\begin{array}{l} \left\{ \begin{array}{l} \text{;RIO} \\ \text{;NORIO} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{;STD} \\ \text{;MSG} \\ \text{;CIR} \end{array} \right\} \end{array} \right] \end{array} \right] \end{array} \right] \end{array} \right] \end{array} \right] \end{array} \right]$$
Parameters for *option*

- recsize* Record size. A positive number indicates words, while a negative number indicates bytes for new files only. For fixed length files, this is the logical record size. For undefined length files, this is the maximum record size. For variable length files, this is the maximum logical record size if the *blockfactor* is 1. If not, this is used to calculate the maximum logical record size and physical record size.
- Records always begin on word boundaries, thus the record size is rounded up to the nearest word boundary for block size calculations. For a binary file or a variable length ASCII file, odd byte lengths are rounded up and the extra byte is available for data.
- However, if a record size with an odd byte length is specified for a fixed or undefined length record file, the extra byte is not available for data. The default is the width of the configured physical record for the associated device. If you do not use the **DEV=** parameter, the default will be **DISC** with 128 records.
- For example, a fixed length ASCII file with a record size specified as 11 bytes has only 11 bytes available for data in each logical record. However, to determine the actual block size, 12 bytes are used for the record size: the block size equals 12 bytes multiplied by the *blockfactor*. If the file is specified as a binary file, the 11 bytes are rounded up to 12 bytes (6 words), all of which are available for each logical record.
- blockfactor* Number of logical records per physical block, for new files only. The default is calculated by dividing the specified *recsize* into the configured block size; this value is rounded downward to an integer that is never less than 1. For variable length record files, the *blockfactor* is always set to 1 after using the original value along with *recsize* to calculate maximum logical record size and physical record size. The *blockfactor* is ignored for undefined length records. The maximum size is 255.

FILE

F, U, or V	Defines the format of the records of the file. A file may contain fixed length records (F), undefined length records (U), or variable length records (V). The default is F for disk files.
BINARY or ASCII	Indicates the type of records. BINARY indicates binary coded records and is the default. ASCII indicates ASCII coded records.
<i>density</i>	Corresponds to tape densities in BPI (bits-per-inch) for new files only. This parameter is only applicable when writing to a tape mounted on the HP 7976A or HP 7978A variable-density tape drive. The density value from a file equation takes precedence over the density specified in FOPEN. The supported densities are 800, 1600 and 6250. For details on the operation of density selection, refer to FOPEN in the <i>MPE V Intrinsic Reference Manual</i> (32022-90007).
<i>numrec</i>	Maximum number of logical records, for new files only. For fixed and undefined length files the maximum value allowed for this field is 2,147,483,647. However, the maximum number of sectors per file is 2,097,120, based on the maximum of 65535 sectors per extent, 32 extents maximum. Thus the actual maximum number of records will be limited by block size (determined by the <i>recsize</i> and <i>blockfactor</i>). An approximate practical limit for <i>numrec</i> is 2,097,119 for files with variable undefined lengths, and 267,382,000 for fixed length files. The default is 1023.
Note 	The file system uses these values to compute other characteristics of the file as well. Therefore, the values (or default values) specified in the FILE command may be valid within their respective fields, but may cause overflow errors in the computation of internally required file specifications. Refer to the <i>MPE File System Reference Manual</i> (30000-90236) for information on calculating file space.
<i>numextents</i>	Maximum number of disk extents. This is a value from 1 to 32. The default is 8.
<i>initialloc</i>	Number of extents to be initially allocated to the file at the time it is opened. This is a value from 1 to 32. The default is 1.
<i>filecode</i>	Code indicating a specially formatted file. This code is recorded in the file label and is available to processes accessing the file through the FGETINFO intrinsic. For this parameter, any user can specify a positive integer ranging from 0 to 32767 or a mnemonic name. Certain integers and mnemonics have been reserved for particular HP-defined meanings, as follows:

Integer	Mnemonic	Meaning
1024	USL	User Subprogram Library
1025	BASD	BASIC Data
1026	BASP	BASIC Program
1027	BASFP	BASIC Fast Program
1028	RL	Relocatable Library
1029	PROG	Program file
1030	NMPRG	Native Mode Program file
1031	SL	Segmented Library
1032	NMSL	Native Mode Executable Library file
1033	NMRL	Native Mode Relocatable Library file
1035	VFORM	View Form file
1036	VFAST	View Fast Forms file
1037	VREF	View Reformat file
1040	XLSAV	Cross Loader ASCII file (SAVE)
1041	XLBIN	Cross Loader Relocated Binary file
1042	XLDSP	Cross Loader ASCII file (DISPLAY)
1050	EDITQ	Edit Quick file
1051	EDTCQ	Edit KEEPQ file (COBOL)
1052	EDTCT	Edit TEXT file (COBOL)
1054	TDPDT	TDP Diary file
1055	TDPQM	TDP Proof Marked QMARKED
1056	TDPP	TDP Proof marked non-COBOL file
1057	TDPCP	TDP Proof Marked COBOL file
1058	TDPQ	TDP Workfile
1059	TDPXQ	TDP Workfile (COBOL)
1060	RJEPN	RJE Punch file
1070	QPROC	QUERY Procedure file
1080	KSAMK	KSAM Key file
1083	GRAPH	GRAPH Specification file
1084	SD	Self-describing file
1090	LOG	User Logging logfile
1100	WDOC	HPWORD Document
1101	WDICT	HPWORD Hyphenation dictionary
1102	WCONF	HPWORD Configuration file
1103	W2601	HPWORD Attended Printer Environment
1110	PCELL	IFS/3000 Character Cell file
1111	PFORM	IFS/3000 Form file
1112	PENV	IFS/3000 Environment file
1113	PCCMP	IFS/3000 Compiled Character Cell file
1114	RASTR	Graphics Image in RASTR Format
1130	OPTLF	OPT/3000 logfile
1131	TEPES	TEPE/3000 Script file
1132	TEPEL	TEPE/3000 logfile
1133	SAMPL	APS/3000 logfile

FILE

Integer	Mnemonic	Meaning
1139	MPEDL	MPEDCP/DRP logfile
1140	TSR	HPToolset Root file
1141	TSD	HPToolset Data file
1145	DRAW	Drawing File for HPDRAW
1146	FIG	Figure File for HPDRAW
1147	FONT	Reserved
1148	COLOR	Reserved
1149	D48	Reserved
1152	SLATE	Compressed SLATE file
1153	SLATW	Expanded SLATE workfile
1156	DSTOR	Store file for RAPID/3000 Utility DICTDBU
1157	TCODE	Code file for Transact/3000 Compiler
1158	RCODE	Code file for Report/3000 Compiler
1159	ICODE	Code file for Inform/3000 Compiler
1166	MDIST	HPDESK Distribution list
1167	MTEXT	HPDESK Text
1168	MARPA	ARPA Message file
1169	MARPD	ARPA Distribution list
1170	MCMND	HPDESK Abbreviated Commands file
1171	MFRTH	Reserved
1172		Reserved
1173	MEFT	Reserved
1174	MCRPT	Reserved
1175	MSERL	Reserved
1176	UCSF	Reserved
1177	TTYPT	Term Type file
1178	TVFC	Term Vertical Format Control file
1192	NCONF	Network Configuration file
1193	NTRAC	Network Trace file
1194	NLOG	Network logfile
1195	MIDAS	Reserved
1211	ANODE	Reserved
1212	INODE	Reserved
1213	INVRT	Reserved
1214	EXCEP	Reserved
1215	TAXON	Reserved
1216	QUERF	Reserved
1217	DOCDR	Reserved
1226	VC	VC file
1227	DIF	DIF file
1228	LANGD	Language Definition file
1229	CHARD	Character Set Definition File
1230	MGCAT	Formatted Application Message Catalog
1236	BMAP	Reserved
1242	BDATA	Basic Data file
1243	BFORM	Basic Field Order File for VPLUS

Integer	Mnemonic	Meaning
1244	BSAVE	Basic Saved Program file
1245	BCNFG	Config. File for Default Option Basic Program
1246	BKEY	Function Key Definition file
1258	PFSTA	Pathflow STATIC file
1259	PFDYN	Pathflow DYNAMIC file
1270	RTDCA	Revisable Form DCA Document*
1271	FFDCA	Document Interchange Unit file HPWORD/150
1272	DIU	DISOSS Filing Information file
1273	PDOC	Search Result Information file
1275	DFI	Reserved
1276	SRI	HPMAP/3000 Map Specification file
1401	CWPTX	Reserved
1421	MAP	Reserved
1422	GAL	HPBRW Dictionary file
1425	TTX	HPBRW Specification file
1428	RDIC	HPBRW Specification file (frozen)
1429	RSPEC	HPBRW Execution file
1430	RSPCF	HPBRW Report Job file
1431	REXEC	HPBRW Intermediate Report file
1432	RJOB	HPBRW Dictionary Output file
1433	ROUTI	HPBRW Print file
1434	ROUTD	HPBRW Configuration file
1435	PRINT	Reserved
1436	RCONF	Native Mode Object file
1441	PIF	PASCAL Source Library
1461	NMOBJ	Reserved
1462	PASLB	PASCAL Source Library
3333		Reserved

* DCA = Document Content Architecture

The default is the unreserved file code of 0.

RIO or NORIO Creates a relative or nonrelative I/O file. **RIO** creates a relative I/O file. The record length parameter will be implicitly changed to fixed record length. **RIO** is a special file access method primarily intended for use by COBOLII programs; however, you can access these files with programs written in any language. **NORIO** creates a nonrelative I/O file. The default is **NORIO**.

RIO and **NORIO** specifications affect only the physical characteristics of the file. If **NOBUF** is specified in the **FILE** command, the file will be accessed in non-**RIO** mode; otherwise **RIO** access is used with **RIO** files. **NOBUF** access is provided for special operations on **RIO** files, such as replicating a **RIO** file. **NOBUF** is not normally used by the **RIO** user. Refer to the *MPE V Intrinsics Reference Manual* (32033-90007) for a discussion of relative I/O.

FILE

STD,MSG,or CIR Defines the type of file. **STD** is a standard MPE V/E disk file. A **MSG** (message) file allows communication between any set of processes. **MSG** acts like a FIFO (first in, first out) queue where records are read from the start of the file and logically deleted and/or are appended to the end of file. **CIR** acts as a normal sequential file until full. When full, the first physical block will be deleted when the next record is written, and remaining blocks will be logically shifted to the front of the file. **CIR** cannot be simultaneously accessed by readers and writers. The default is **STD**.

Syntax for *access*

$$\left[\left\{ \begin{array}{l} ;\text{NOCCTL} \\ ;\text{CCTL} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} ;\text{NOMULTI} \\ ;\text{MULTI} \\ ;\text{GMULTI} \end{array} \right\} \right] \left[;\text{ACC} = \left\{ \begin{array}{l} \text{IN} \\ \text{OUT} \\ \text{UPDATE} \\ \text{OUTKEEP} \\ \text{APPEND} \\ \text{INOUT} \end{array} \right\} \right] \\ \left[\left\{ \begin{array}{l} ;\text{WAIT} \\ ;\text{NOWAIT} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} ;\text{NOMR} \\ ;\text{MR} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} ;\text{BUF} = [\text{numbuffers}] \\ ;\text{NOBUF} \end{array} \right\} \right] \\ \left[\left\{ \begin{array}{l} ;\text{LOCK} \\ ;\text{NOLOCK} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} ;\text{EXC} \\ ;\text{SHR} \\ ;\text{EAR} \\ ;\text{SEMI} \end{array} \right\} \right] \left[\left\{ \begin{array}{l} ;\text{COPY} \\ ;\text{NOCOPY} \end{array} \right\} \right] [;\text{FORMS} = \text{formsmg}] \\ \left[\left\{ \begin{array}{l} ;\text{NOLABEL} \\ ;\text{LABEL} = [\text{volsetid}] \left[, \left\{ \begin{array}{l} \text{IBM} \\ \text{ANS} \end{array} \right\} \right] [, [\text{expdate}] [, \text{seq}]] \end{array} \right\} \right]$$

Parameters for *access*

NOCCTL or CCTL	Indicates whether carriage control characters are specified or not. NOCCTL indicates that carriage control characters are not specified in "writes" to the file. CCTL indicates that carriage control characters are supplied in "writes" to the file. The default is NOCCTL .
IN,OUT, UPDATE, OUTKEEP, APPEND, or INOUT	Defines the type of file access. IN only permits READ access to the file and is the default for input devices. OUT only permits WRITE access to the file and is the default for output devices. UPDATE permits any type of access to the file. OUTKEEP only permits WRITE access to the file, but previous data is not deleted. APPEND only permits APPEND access to any file. INOUT only permits input/output access; any file intrinsic except FUPDATE can be issued against the file.

Note



You may not use **APPEND** access with labeled tapes. Instead, use the **ADDF** or **NEXT** options of the **;LABEL** parameter.

BUF= <i>numbuffers</i> or NOBUF	Specifies whether buffers are to be allocated to the file. The <i>numbuffers</i> parameter is the number of buffers (1 to 16) to be allocated for the file. The <i>numbuffers</i> parameter is ignored for terminals. The default is 2 buffers. NOBUF specifies that no buffers are allocated for the file.						
EXC,SHR, EAR,or SEMI	Specifies shared or exclusive file access. EXC is exclusive access; after the file is opened no other accessors are permitted. For message and circular files, EXC means one writer and one reader. SHR is shared access; after the file is opened other accessors are permitted. EAR and SEMI are identical. For message files they allow one exclusive reader and multiple writers. For non-message files they allow one exclusive writer and multiple readers. The default is EXC , except with READ -only file access (IN).						
NOLABEL or LABEL	Specifies whether this tape is labeled or unlabeled. NOLABEL specifies an unlabeled tape. LABEL specifies a labeled tape. The default is NOLABEL .						
<i>volsetid</i>	Up to six alphanumeric characters identifying a labeled magnetic tape volume set.						
ANS or IBM	Type of standard label. ANS is an ANSI standard label. IBM is an IBM standard label. The default is ANS .						
<i>expdate</i>	Month, day, year, written in the format <i>mm/dd/yy</i> . This specifies the expiration date of the file, or the date after which information contained in the file is no longer useful. The file can be overwritten without operator reconfirmation after this date. The default is 00/00/00: the file can be overwritten immediately.						
<i>seq</i>	Either an absolute file number between 1 and 9999 (inclusive), or one of the following, which specifies the position of the file relative to other files on the tape: <table> <tr> <td>0</td><td>Causes a search of all volumes until the file is found.</td></tr> <tr> <td>ADDF</td><td>ADDF positions the tape to add a new file onto the end of the volume (or last volume in a multi-volume set). You should not use ADDF to add to a new labeled tape volume.</td></tr> <tr> <td>NEXT</td><td>NEXT positions the tape at the next file on the tape. If this is the first FOPEN, then NEXT will cause the tape to be positioned at the first file on the tape. If the previous FCLOSE specified REWIND, the tape backspaces to the last file and remains positioned as it was, at the previous file. This is the default. Refer to FOPEN in section II of the <i>MPE V Intrinsic Reference Manual</i> (32033-90007) for more information.</td></tr> </table>	0	Causes a search of all volumes until the file is found.	ADDF	ADDF positions the tape to add a new file onto the end of the volume (or last volume in a multi-volume set). You should not use ADDF to add to a new labeled tape volume.	NEXT	NEXT positions the tape at the next file on the tape. If this is the first FOPEN , then NEXT will cause the tape to be positioned at the first file on the tape. If the previous FCLOSE specified REWIND , the tape backspaces to the last file and remains positioned as it was, at the previous file. This is the default. Refer to FOPEN in section II of the <i>MPE V Intrinsic Reference Manual</i> (32033-90007) for more information.
0	Causes a search of all volumes until the file is found.						
ADDF	ADDF positions the tape to add a new file onto the end of the volume (or last volume in a multi-volume set). You should not use ADDF to add to a new labeled tape volume.						
NEXT	NEXT positions the tape at the next file on the tape. If this is the first FOPEN , then NEXT will cause the tape to be positioned at the first file on the tape. If the previous FCLOSE specified REWIND , the tape backspaces to the last file and remains positioned as it was, at the previous file. This is the default. Refer to FOPEN in section II of the <i>MPE V Intrinsic Reference Manual</i> (32033-90007) for more information.						

FILE

NOMULTI, MULTI, or GMULTI	Specifies the sharing of files in jobs and sessions. NOMULTI prohibits the sharing of files in MULTI mode and is the default. MULTI allows users to access the file concurrently and treat the file as if no buffering is taking place. Access control information can be shared by processes in the same CI process tree (that is, parent-to-child processes) with MULTI . GMULTI is the same as MULTI , except that it allows users access from different jobs and/or sessions.
NOMR or MR	Specifies whether multirecord access is permitted. NOMR specifies that no multirecord access is permitted. MR allows multirecord access to the file. The default is NOMR .
WAIT or NOWAIT	Specifies whether I/O requests are to be completed or queued before control returns to the program. WAIT completes I/O requests to the file before control is returned to the program. NOWAIT returns control to the program as soon as I/O requests are queued by MPE V/E; only privileged mode programs are allowed. The program thus does not have to wait for the physical I/O to be complete before resuming execution. This parameter also implies NOBUF .
NOLOCK or LOCK	Specifies whether dynamic locking or unlocking is to be permitted. NOLOCK prohibits dynamic locking or unlocking of a file through the FLOCK/FUNLOCK intrinsics. LOCK allows dynamic locking or unlocking through the FLOCK/FUNLOCK intrinsics. The default is NOLOCK .
COPY or NOCOPY	Specifies whether files can be copied. COPY allows MSG , KSAM , and CIR files to be either copied (logical data record READ) or replicated (block READ and WRITE , completely duplicating the file) to another file. NOCOPY accesses the file in its natural mode, that is, as a MSG file. The default is NOCOPY .
<i>formsmsg</i>	A message to the operator requesting that certain forms be mounted. The message must be displayed and verified before the output data can be printed on a line printer. The message is a string of no more than 49 ASCII characters terminated by a period. You may send control characters for bells and inverse video to the system console using this parameter. Attempts to send other control characters, however, will result in a display of blanks and the associated control character letter when the forms message appears on the system console. The default is "no forms message sent".

Syntax for *disposition*

$$\left[\left\{ \begin{array}{l} ;\text{DEL} \\ ;\text{TEMP} \\ ;\text{SAVE} \end{array} \right\} \right]$$
**Parameters for *disposition***

- DEL** The file is deleted when closed.
- SAVE** The file is saved in the permanent file domain when closed.
- TEMP** The file is saved in the job/session temporary domain when closed.

If none of these parameters are supplied, the disposition of the file is as it was when opened, or as specified by the **FCLOSE** intrinsic call that your program issued. Refer to the *MPE File System Reference Manual* (30000-90236) for more information on file domains.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **FILE** command allows you to change the specifications for files at run-time, as well as the devices on which they reside, thus overriding specifications supplied via the **FOPEN** intrinsic. The **FILE** command remains in effect for the entire job or session unless revoked by the **RESET** command or superseded by another **FILE** command.

To use the **FILE** command for a file, you must have a valid, formal file designator (the name by which your program recognizes the file). The formal file designator provides a way for commands and code outside your program to reference the file.

Note

When you are programming in a language other than SPL, and you did not write the **FOPEN** intrinsic calls for files used by your program, the only way you can control or change the programmatic file specifications is through the **FILE** command. (Refer to the reference manual of the language you are using for more details.)

FILE

Examples

A program called **MYPROG** references two files by the file names (*formal designators*) **SOURCE** and **DEST**, but you wish to use two existing disk files **INX** and **OUTX** as the actual files for the program. To do this, enter:

```
:FILE SOURCE=INX  
:FILE DEST=OUTX  
:RUN MYPROG
```

Now you wish to run the program using input from a card reader with the device class name **CARD**, which has 80-byte records:

```
:FILE SOURCE,OLD;REC=-80;DEV=CARD
```

You want the output sent to a new file called **FILEX**, with 64-word fixed length records, blocked two records per block in ASCII code. **FILEX** is limited to 800 records among 10 extents, two of which are to be allocated immediately. You want the file permanently saved when **MYPROG** closes it. Enter:

```
:FILE DEST=FILEX,NEW;REC=64,2,F,ASCII;DISC=800,10,2;SAVE  
:RUN MYPROG
```

Note



The file equation only modifies those items specified. All other attributes come from the parameters (or the defaults, where parameters are omitted) specified in the **FOPEN** call for the file **DEST**.

Implicit FILE Commands for Subsystems

When an actual file designator appears as a command parameter, it is automatically equated to a formal file designator. An implicit FILE command issued by the command executor then uses the formal file designator within the subsystem. For example, within the FORTRAN compiler, the formal file designator for the text file input is FTNTEXT. Suppose you specify a file named ALSFILE for text file input as shown below:

```
:FORTRAN ALSFILE
```

MPE V/E implicitly issues the following FILE command, invisible to you:

```
:FILE FTNTEXT=ALSFILE
```

You cannot backreference any of the formal file designators associated with the command as actual file designators. Therefore, do not, use the formal file designators FTNTEXT, FTNUSL, FTNLIST, FTNMAST or FTNNEW as actual file names. The use of FTNTEXT as a file name, as in the following example, is invalid because the implicit FILE command issued by the FORTRAN compiler will then backreference itself:

```
:FORTRAN;*FTNTEXT  
:FILE FTNTEXT=*FTNTEXT
```

The following example shows the correct use of **formal designator*, in this case specifying a file on magnetic tape used as a source file during a FORTRAN compilation:

```
:FILE SOURCE=TAPE1,OLD;DEV=TAPE;REC=-80  
:FORTRAN;*SOURCE
```

The command executor issues the following implicit FILE command, backreferencing your previous FILE command:

```
:FILE FTNTEXT=*SOURCE
```

Implicit FILE commands, like explicit FILE commands, cancel any previous FILE commands that reference the same formal file designators. Formal file designators are described in each compiler command description throughout this manual.

Additional Discussion

MPE V Intrinsics Reference Manual (32033-90007)
MPE File System Reference Manual (30000-90236)

FOREIGN

Directs MPE V/E to treat the disk volume mounted on the specified logical device as foreign.

Syntax

FOREIGN *ldev*

Parameter

ldev The logical device number of a disk drive.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may issue it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The **FOREIGN** command forces MPE V/E to treat the volume currently mounted on the logical device as a foreign disk. The device class associated with the logical device must be foreign and the drive must be online, but the volume cannot be in use.

Caution



If the restrictions stated above are met, the system will treat the logical device as a foreign disk. As long as the volume is not a system domain disk, MPE V/E performs no security checking. Use caution with this command, because any disk (system, private volume, etc.) could be mounted and inadvertently treated as a foreign disk. Since the foreign disk facility may write on the MPE V/E label area, file labels could be destroyed and important information lost.

Examples

To direct MPE V/E to treat the disk mounted on logical device #3 as foreign, first verify the status of the disk by entering:

:DSTAT 3

LDEV-TYPE -----	STATUS -----	VOLUME (VOLUME SET-GEN) -----
3-7902	SERIAL	*UNALLOCATED*

Then enter:

:FOREIGN 3

To verify the change, enter:

:DSTAT 3

LDEV-TYPE -----	STATUS -----	VOLUME (VOLUME SET-GEN) -----
3-7902	FOREIGN	*UNALLOCATED*

FORTGO

Compiles, prepares, and executes a FORTRAN/3000 program. FORTRAN/3000 is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
FORTGO [ textfile ] [ , [ listfile ] [ , [ masterfile ] [ , newfile ] ] ] [ ;INFO=quotedstring ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is FTNTEXT. The default is \$STDIN.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is FTNLIST. The default is \$STDLIST.
<i>masterfile</i>	Actual file designator of the master file with which <i>textfile</i> is merged to produce a composite source. This can be any ASCII input file. The formal file designator is FTNMAST. The default is that the file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the file resulting when <i>textfile</i> and <i>masterfile</i> are merged. This can be any ASCII output file. The formal file designator is FTNNEW. The default is that the file is not written.

Note



The formal file designators (FTNTEXT, FTNLIST, FTNMAST, and FTNNEW) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the MPE V/E FILE command.

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

Operation

The **FORTGO** command compiles, prepares, and executes a FORTRAN/3000 program. If you do not specify a source file, MPE V/E expects input from your standard input device. If you do not specify *listfile*, MPE V/E writes the listing to your standard output device.

The USL file created during the compilation is a system-defined temporary file called \$OLDPASS, which is passed directly to the Segmenter and cannot be accessed.

Examples

To compile, prepare, and execute a FORTRAN/3000 program entered from the disk file **SOURCE** and transmit the resulting program listing to the disk file **LISTFL**, enter:

```
:FORTGO SOURCE,LISTFL
```

To enter your source input from a device other than your standard input device, and/or direct the listing to a device other than your standard list device, simply name the input and listing files as command parameters. In the example below, the source listing is read from magnetic tape, formally identified by the file name **MTAPE**. Output is sent to the printer, identified by the file name **PRTR**.

```
:FILE MTAPE;DEV=TAPE  
:FILE PRTR;DEV=FASTLP
```

MTAPE and **PRTR** are then backreferenced in the **FORTGO** command, as shown here:

```
:FORTGO *MTAPE,*PRTR
```

Additional Discussion

FORTRAN/3000 Reference Manual (30000-90040)

MPE Segmenter Reference Manual (30000-90011)

FORTPREP

Compiles and prepares a FORTRAN/3000 program. FORTRAN/3000 is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
FORTPREP [ textfile ] [ , [ progfile ] [ , [ listfile ] [ , [ masterfile ] [ , newfile ] ] ] ]  
[ ;INFO=quotedstring ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is FTNTEXT . The default is \$STDIN .
<i>progfile</i>	Actual file designator of the program file to which the prepared program segments are written. You get the best results by omitting the <i>progfile</i> parameter. When you omit <i>progfile</i> , the MPE V/E Segmenter creates the program file, which resides in the temporary file domain as \$OLDPASS . If you create your own program file, you must do so in one of two ways: ■ Use the MPE V/E BUILD command, and specify a file code of 1029 or PROG, and a <i>numextents</i> value of 1. The PREP command then uses this file. ■ Specify a nonexistent file in the <i>progfile</i> parameter, resulting in the creation of a job/session temporary file of the correct type.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is FTNLIST . The default is \$STDLIST .
<i>masterfile</i>	Actual file designator of the master file with which <i>textfile</i> is merged to produce a composite source. This can be any ASCII input file. The formal file designator is FTNMAST . The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the file resulting when <i>textfile</i> and <i>masterfile</i> are merged. This can be any ASCII output file. The formal file designator is FTNNEW . The default is that the file is not written.

Note



The formal file designators (**FTNTEXT**, **FTNLIST**, **FTNMAST**, and **FTNNEW**) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the MPE V/E **FILE** command.

Use

You may issue this command from a session or a job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **FORTPREP** command compiles and prepares a FORTRAN/3000 program into a program file stored on disk. If you do not specify a source file, MPE V/E expects the input from your standard input device. If you do not specify a *listfile*, MPE V/E sends the output to your standard list device.

The USL (user subprogram library) file created during compilation is a system-defined temporary file **\$OLDPASS**, which is passed directly to the Segmenter. The Segmenter also uses the file **\$OLDPASS**. The prepared program segments are written to it, thus overwriting any existing temporary file of that name.

If you have no need to examine the USL file, use the default for *progfile*. This way, MPE V/E will create a program file for you, ensuring the best results. If, on the other hand, you want to store the USL file and the program file as separate entities, create a permanent program file with the **BUILD** command, and reference it in the **FORTPREP** command line. When you create the file, you must specify **PROG** or **1029** as the file code, and a *numextents* parameter value of **1**, as shown in the example below.

```
:BUILD PROGX;CODE=PROG;DISC=1
```

Examples

To compile and prepare a FORTRAN/3000 program entered from your standard input device, into the standard default file **\$OLDPASS**, with the listing printed on your standard list device, enter:

```
:FORTPREP
```

To compile and prepare a FORTRAN/3000 source program from a text file named **TEXTX** into a program with the file name **PROGX**, and send the resulting listing to the list file **LISTX**, enter:

```
:FORTPREP TEXTX,PROGX,LISTX
```

The **FORTPREP** command combines the compilation and preparation steps. The compiled program segments, stored in the file **\$OLDPASS**, are prepared and stored in the program file **PROGX**. Therefore, it is equivalent to:

```
:FORTRAN TEXTX, LISTX  
:PREP $OLDPASS,PROGX
```

Additional Discussion

FORTRAN/3000 Reference Manual (30000-90040)
MPE Segmenter Reference Manual (30000-90011)

FORTRAN

Compiles a FORTRAN/3000 program. FORTRAN/3000 is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
FORTRAN [ textfile ] [ , [ uslfile ] [ , [ listfile ] [ , [ masterfile ] [ , newfile ] ] ] ]  
[ ; INFO=quotedstring ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is FTNTEXT . The default is \$STDIN .
<i>uslfile</i>	Actual file designator of the USL (user subprogram library) file to which the object program is written, which can be any binary output file with file code of USL or 1024. The formal file designator is FTNUSL. If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS. If you use the <i>uslfile</i> parameter, you must create the file in one of four ways: ■ By using the MPE V/E SAVE command to save the default USL file \$OLDPASS created during a previous compilation. ■ By building the USL file with the Segmenter -BUILDUSL command. (Refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011).) ■ By creating a new USL file with the MPE V/E BUILD command and specifying a file code of USL or 1024. ■ By specifying a nonexistent <i>uslfile</i> parameter, thereby creating a permanent file of the correct size and type.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is FTNLIST . The default is \$STDLIST .
<i>masterfile</i>	Actual file designator of the master file with which <i>textfile</i> is merged to produce a composite source. This can be any ASCII input file. The formal file designator is FTNMAST . The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified.
<i>newfile</i>	Actual file designator of the merged <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. The formal file designator is FTNNEW . The default is that no file is written.

Note



The formal file designators (**FTNTEXT**, **FTNUSL**, **FTNLIST**, **FTNMAST**, and **FTNNEW**) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the MPE V/E **FILE** command.

Use

You may issue this command from a session or job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **FORTRAN** command compiles a FORTRAN/3000 program into a USL file on disk. If you do not specify *textfile*, MPE V/E expects input from your standard input device. If you do not specify *listfile*, MPE V/E sends the listing to your standard list device.

If you create the USL file (using the MPE V/E **BUILD** command) before compiling the program, you must assign it a file code of USL or 1024. If you omit this parameter, the compiled program segments will be stored in the temporary file **\$OLDPASS**.

Examples

To compile a FORTRAN/3000 program entered from your standard input device into an object program in the USL file **\$OLDPASS**, and write the listing to your standard list device, enter:

```
:FORTRAN
```

The following example compiles a program from the source file **MYSOURCE** and stores the object code into the USL file **MYUSL**. The program listing is stored in the disk file **MYLIST**.

```
:FORTRAN MYSOURCE,MYUSL,MYLIST;INFO= "$CONTROL BOUNDS"
```

To compile a FORTRAN/3000 program and store the object code into a USL file you create with the **BUILD** command, enter:

```
:BUILD OBJECT;CODE=USL          ** You must specify the CODE parameter  
:FORTRAN SOURCE,OBJECT,LISTFL  ** as chosen to create a USL file with the  
                                ** BUILD command. **
```

Additional Discussion

FORTRAN/3000 Reference Manual (30000-90040)

MPE Segmenter Reference Manual (30000-90011)

FREERIN

Releases a global resource identification number (RIN).

Syntax

FREERIN *rin*

Parameters

rin The resource identification number (RIN) to be released. It must be a number from 1 to the configured maximum.

Use

You may issue this command from a session or job. You may not use it from a program or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **FREERIN** command releases a resource identification number. A resource identification number is used to manage a resource shared among two or more jobs or sessions so that only one job or session at a time can access that resource.

You acquire a RIN from the system by entering the **GETRIN** command. When all users are finished with the RIN, the user who acquired it returns it to the system by entering the **FREERIN** command. To free a RIN, you must be the original owner of that RIN, that is, the user who actually issued the **GETRIN** command that allocated the RIN and assigned it a password.

Example

To release RIN #6, enter:

:FREERIN 6

Additional Discussion

MPE V Intrinsic Reference Manual (32033-90007)

FTN

Compiles a FORTRAN 77 program. FORTRAN 77 is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
FTN [ textfile ] [ , [ uslfile ] [ , listfile ] ] [ ; INFO=quotedstring ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is FTNTEXT . The default is \$STDIN .
<i>uslfile</i>	Actual file designator of the USL file to which the object code is stored, which can be any binary output file with a file code of USL or 1024. Its formal file designator is FTNUSL . If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS . If entered, this parameter indicates that the USL file was created in one of four ways: ■ By using the MPE V/E SAVE command to save the default USL file \$OLDPASS created during a previous compilation. ■ By building the USL file with the MPE Segmenter -BUILDUSL command. Refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011). ■ By creating a new USL file specifying the MPE V/E BUILD command and specifying a file code of USL or 1024. ■ By specifying a nonexistent <i>uslfile</i> parameter, thereby creating a permanent file of the correct size and type.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is FTNLIST . The default is \$STDLIST .

Note



The formal file designators used in this command (**FTNTEXT**, **FTNUSL**, and **FTNLIST**) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on "Implicit FILE Commands For Subsystems" under the MPE V/E **FILE** command.

<i>quotedstring</i>	A sequence of characters between two single quotation marks or between two double quotation marks. You may use the delimiter as part of the string as long as the delimiter appears twice. Any occurrence of two single quotation marks in a row, or two double quotes in a row, is considered part of the string, and not the terminating delimiter. The INFO=quotedstring parameter is used in the FORTRAN 77 programming language to pass initial compiler options to a program. For more information, refer to the RUN command in this chapter.
---------------------	--

FTN

Use

You may issue this command from a session or a job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **FTN** command compiles a FORTRAN 77 program and stores the object code in a user subprogram library (USL) file on disk. If *textfile* is not specified, MPE V/E expects the source program to be entered from your standard input device. If you do not specify a *listfile*, MPE V/E sends the program listing to your standard list device and identifies it by the formal file designator, FTNLIST.

If you create the USL file prior to compilation, you must specify a file code of USL or 1024. If you omit the *uslfile* parameter, the object code is saved in the temporary file domain as \$OLDPASS. To keep it as a permanent file, you must save \$OLDPASS under another name.

Examples

The following example compiles a FORTRAN 77 program entered from your standard input device and stores the object program in the USL file \$OLDPASS. The listing is then sent to your standard list device.

```
:FTN
```

The next example compiles a FORTRAN 77 program contained in the disk file FORTSRC, and stores the object program in the USL file FORTOBJ. The program listing is stored in the disk file LISTFILE.

```
:BUILD FORTOBJ;CODE=USL  
:FTN FORTSRC,FORTOBJ,LISTFILE
```

Additional Discussion

FORTRAN 77 Reference Manual (5957-4685)

MPE Segmenter Reference Manual (30000-90011)

FTNGO

Compiles, prepares, and executes a FORTRAN 77 program. FORTRAN 77 is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
FTNGO [textfile] [ ,listfile] [ ;INFO=quotedstring]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is FTNTEXT . The default is \$STDLIST .
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is FTNLIST . The default is \$STDLIST .

Note



The formal file designators (**FTNTEXT** and **FTNLIST**) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the MPE V/E **FILE** command.

<i>quotedstring</i>	A sequence of characters between two single quotation marks or between two double quotation marks. You may use the delimiter as part of the string so long as the delimiter appears twice. Any occurrence of two single quotation marks in a row, or two double quotes in a row, is considered part of the string, and not the terminating delimiter. The INFO=quotedstring parameter is used in the FORTRAN 77 programming language to pass initial compiler options to a program. For more information, refer to the RUN command in this chapter.
---------------------	---

Use

You may issue this command from a session or job. You may not use it from a program, or in **BREAK**. Pressing **(BREAK)** suspends execution.

Operation

The **FTNGO** command compiles, prepares, and executes a FORTRAN 77 program. If *textfile* is omitted, MPE V/E expects input from your standard input device. If you do not specify *listfile*, MPE V/E sends the program listing to the formal file designator **FTNLIST**. The default is **\$STDLIST**.

The USL file created during the compilation is the system-defined temporary file **\$OLDPASS**, which is passed directly to the Segmenter. It cannot be accessed because the Segmenter also uses **\$OLDPASS** to store the prepared program segments, overwriting any existing temporary file of the same name.

FTNGO

Examples

To compile, prepare, and execute a FORTRAN 77 program entered from your standard input device, with the program listing sent to your standard list device, enter:

:FTNGO

To compile, prepare, and execute a FORTRAN 77 program from the disk file FORTSRC and send the program listing to the file LISTFILE, enter:

:FTNGO FORTSRC,LISTFILE

Additional Discussion

FORTRAN 77 Reference Manual (5957-4685)

MPE Segmenter Reference Manual (30000-90011)

FTNPREP

Compiles and prepares a FORTRAN 77 program. FORTRAN 77 is not part of the HP 3000 Fundamental Operating System and must be purchased separately.

Syntax

```
FTNPREP [ textfile ] , [ progfile ] [ , listfile ] [ ; INFO=quotedstring ]
```

Parameters

textfile Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is **FTNTEXT**. The default is **\$STDIN**.

progfile Actual file designator of the program file to which the prepared program segments are written. You get the best results by omitting the *progfile* parameter. When you omit *progfile*, the MPE Segmenter creates the program file, which is stored in the temporary file domain as **\$OLDPASS**. If you do create your own program file, you must do so in one of two ways:

- By using the MPE V/E **BUILD** command and specifying a file code of 1029 or **PROG**, and a *numextents* value of 1. The **PREP** command then uses this file.
- By specifying a nonexistent file in the *progfile* parameter, in which case a job/session temporary file of the correct size and type is created.

listfile Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is **FTNLIST**. The default is **\$STDLIST**.

Note



The formal file designators (**FTNTEXT**, **FTNPROG**, and **FTNLIST**) used in this command cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the MPE V/E **FILE** command.

quotedstring A sequence of characters between two single quotation marks or between two double quotation marks. You may use the delimiter as part of the string so long as the delimiter appears twice. Any occurrence of two single quotation marks in a row, or two double quotes in a row, is considered part of the string, and not the terminating delimiter.

The *INFO=quotedstring* parameter is used in the FORTRAN 77 programming language to pass initial compiler options to a program. **PASCAL/3000** brackets the *quotedstring* with dollar signs and places it before the first line of source code in the text file. For more information, refer to the **RUN** command in this chapter.

FTNPREP

Use

You may issue this command from a session or job. You may not use it from a program, or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **FTNPREP** command compiles and prepares a FORTRAN 77 program into a program file on disk. If you do not specify *textfile*, MPE V/E expects input from the current input device. If you do not specify *listfile*, MPE V/E sends the listing output to the formal file designator **FTNLIST**. The

default is **\$STDLIST**. The USL file **\$OLDPASS**, created during compilation, is a temporary file passed directly to the Segmenter. You may access it only if you do not use the default for *progfile*. This is because the Segmenter also uses **\$OLDPASS** to store the prepared program segments, overwriting any existing temporary file of the same name.

Examples

The following example compiles and prepares a FORTRAN 77 program entered through your standard input device and stores the prepared program segments in the file **\$OLDPASS**. The listing is printed on your standard list device.

:FTNPREP

To compile and prepare a FORTRAN 77 source program from the source file **FORTSRC**, storing it in **FORTPROG**, and sending the listing to your standard list device, enter:

:FTNPREP FORTSRC,FORTPROG

Additional Discussion

FORTRAN 77 Reference Manual (5957-4685)

MPE Segmenter Reference Manual (30000-90011)

FULLBACKUP

Performs a backup of MPE V/E and all the files on the system.

Syntax

```
FULLBACKUP [ dumptape ] [ , auxlistfile ]
```



Parameters

- dumptape* Specifies the magnetic tape destination for the backup. If *dumptape* is specified, it must be backreferenced to a previous **FILE** command, or the file must be equated to the formal file designator **DUMPTAPE**. The default is **TAPE**. If there is no device class **TAPE**, the device class **CTAPE** is used.
- auxlistfile* Actual file designator of the output device file to which all listings requested during the execution of the backup process are written. The default is **\$STDLIST**.

Use

You may issue this command from a session or job, but not in **BREAK**. Pressing **BREAK** suspends execution. You require system manager (SM) or system supervisor (OP) capability to use it.

Operation

The **FULLBACKUP** command starts the backup process and copies MPE V/E and all the files on the system, including any ACDs attached to the files. The two commands **PARTBACKUP** and **FULLBACKUP** provide an alternative to the **SYSDUMP** dialog. The only user interaction required with these commands is mounting tapes or serial disks and replying to any mount request. These replies are unnecessary, however, if the devices are configured as **AUTO REPLY**.

During the backup, **FULLBACKUP** will display messages at one-minute intervals to report its progress:

```
:STORE OPERATION IS 27% COMPLETE
```

Examples

In the following example, the first **FILE** command defines **DUMP** as a magnetic tape file. The second **FILE** command defines **LIST** as a line printer. The **TAPE** and **LP** parameters are device class names arbitrarily defined during the previous system configuration.

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

GETLOG

Establishes a logging identifier on the system.

Syntax

$$\text{GETLOG } \textit{logid}; \text{LOG}=\textit{logfile} \left[\left\{ \begin{array}{l} ,\text{DISC} \\ ,\text{TAPE} \\ ,\text{SDISC} \\ ,\text{CTAPE} \end{array} \right\} \right] \left[;\text{PASS}=\textit{password} \right] \left[\left\{ \begin{array}{l} ;\text{AUTO} \\ ;\text{NOAUTO} \end{array} \right\} \right]$$

Parameters

<i>logid</i>	The logging identifier to be established. This must contain from one to eight alphanumeric characters beginning with an alphabetic character.
<i>logfile</i>	The name of the file to receive data from the logging procedure. It must contain from one to eight alphanumeric characters, beginning with an alphabetic character. You must also specify the device class on which the <i>logfile</i> resides, that is, DISC, TAPE, SDISC, or CTAPE.
<i>password</i>	Logging identifier password, optionally assigned by the creator for protection against illegal use of a particular identifier. The password must contain from one to eight alphanumeric characters, beginning with an alphabetic character.
AUTO	Initiates an automatic CHANGELOG if the log file becomes full. Refer to CHANGELOG in this chapter.
NOAUTO	Prevents initiation of an automatic CHANGELOG. A CHANGELOG is not performed if the log file is full.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You require system supervisor (OP) or user logging (LG) capability to use it.

Operation

The GETLOG command specifies a logging identifier to be used each time a particular logging process is used. This command is frequently used with data bases, so that each test task you run will write to a logging file. Therefore, data recovery is easier because you know where the task failed.

The creator of the logging identifier must have user logging (LG) capability to execute this command. Other users can be allowed access to this logging identifier by notifying them of the identifier and password. Note that if a password is specified, it is required whenever the logging process is accessed. If you access the logging system with this identifier, you must supply the identifier and password in the OPENLOG intrinsic.

To use the `AUTO` parameter, the log process for *logid* must be enabled for changing. You may do this by ending the log file name with the numeric characters “001” (for example, *fname001*). This naming convention works in conjunction with the file set number to generate sequential file names automatically.

There cannot be two logging identifiers with the same name on the system at the same time. You can use the `LISTLOG` command to find out what logging identifiers currently exist.

Example

To create the logging identifier **FINANCE** and associate it with the disk log file **A**, enter:

```
:GETLOG FINANCE;LOG=A,DISC
```

Additional Discussion

MPE V System Operation and Resource Management Reference Manual(32033-90005)

MPE V Intrinsic Reference Manual (32033-90007)

GETRIN

Acquires a global resource identification number (RIN) and assigns a password to it.

Syntax

`GETRIN rinpassword`

Parameters

rinpassword Password of the intrinsic that locks the RIN. The password must contain from one to eight alphanumeric characters, beginning with an alphabetic character.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **GETRIN** command acquires a global RIN from the MPE V/E RIN pool, usually during a session. You must assign an arbitrary password for the RIN, which aids in restricting its use to authorized users. You can then give this RIN and the associated password to cooperating users so that they can lock and unlock it. For instructions on how to lock and unlock a RIN, and how to pass a RIN and its password as intrinsic parameters, refer to the *MPE V Intrinsics Reference Manual* (32033-90007).

Users who know the RIN and its password can use it in their programs (in jobs or sessions) until the user who acquired the RIN releases it with the **FREERIN** command. The RIN acquired is always a unique, positive integer. The total number of RINs that MPE V/E can allocate is specified when the system is configured, but cannot exceed 1024. If all currently available RINs have been acquired by other users, MPE V/E rejects your request and issues the message:

RIN TABLE FULL

In this case, you must wait until one of the RINs becomes available, or request that your system manager increase the maximum number of RINs that can be assigned.

Example

To acquire a global RIN and assign to it the password **MYRIN**, enter:

`:GETRIN MYRIN`

MPE V/E responds with the RIN number assigned, for example:

RIN: 1

Additional Discussion

MPE V Intrinsics Reference Manual (32033-90007)

GIVE

Assigns a downed device to diagnostics.

Syntax

`GIVE ldev`

Parameters

ldev The logical device number of the downed device.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The **GIVE** command gives control of the device specified by the *ldev* to diagnostics so that no other process can access the device while diagnostic tests are in progress. Before the device can be given to diagnostics, you must first take it offline with the **DOWN** command, as shown in the example below.

Examples

To assign a downed device with logical device #81 to diagnostics, enter:

```
:DOWN 81
:GIVE 81
11:20/3/LDEV#81 IN USE BY DIAGNOSTICS
```

To assign a spooled, job-accepting card reader to diagnostics, enter:

```
:STOPSP00L 5    ** Stops spooling process for ldev #5. **

11:16/31SP#5/STOPPED
11:16/31/LDEV#5 NOT READY

:REFUSE 5        ** Prevents jobs and sessions on the device. **
:ABORTIO 5       ** Aborts pending I/O for the device. **

11:16/3/LDEV#5 NO I/O PENDING

:DOWN 5                ** Takes ldev 5 offline. **
:GIVE 5                ** Give ldev 5 to diagnostics. **

11:16/3/LDEV#5 IN USE BY DIAGNOSTICS
```

HEADOFF

Stops header/trailer output to a device.

Syntax

`HEADOFF ldev`

Parameters

ldev The logical device number of the printer or card punch affected by the command.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The **HEADOFF** command lets you stop header and/or trailer information from being printed. Header and trailer information is data which appears before and after a file, but which is not part of the file's text. It identifies the file by session number, output spool file number, session name (if any), user and account. It also lists the date and time the file was printed.

When the header/trailer facility is enabled and output is directed to a card punch, MPE V/E automatically punches a header card and a trailer card identifying the job. If output is directed to a line printer, MPE V/E automatically prints header and trailer pages identifying the job that produced the file.

If the device is in use and a header has already been printed when you issue the **HEADOFF** command, your request to suppress header/trailer output takes effect after the corresponding trailer is printed.

Example

To stop header/trailer output to logical device #6, enter:

`:HEADOFF 6`

HEADON

Resumes header/trailer output to a device.

Syntax

HEADON *ldev*

Parameters

ldev The logical device number of the printer or card punch affected by the command.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The **HEADON** command lets you resume printing header and/or trailer information again. Header and trailer information is data which appears before and after a file, but which is not part of the file's text. It identifies the file by session number, output spool file number, session name (if any), user and account. It also lists the date and time the file was produced.

When the header/trailer facility is enabled and output is directed to a card punch, MPE V/E automatically punches a header card and a trailer card identifying the job. If output is directed to a line printer, MPE V/E automatically prints header and trailer pages identifying the job that produced the file.

If the device is in use, your request to resume header/trailer output takes effect after the current output is complete.

The header/trailer facility is enabled on all cold loads.

Example

To resume header/trailer output to logical device #6, enter:

:HEADON 6

HELLO

Initiates an interactive session.

Syntax

```
HELLO [ sessionname, ] username [ /userpass ] . acctname [ /acctppass ]  
[ , groupname [ /grouppass ] ] [ ; TIME = cpusecs ] [ ; PRI = { BS  
CS  
DS  
ES } ]  
[ { ; INPRI = inputpriority } ] [ { ; TERM = { termtype  
termname } ]
```

Parameters

<i>sessionname</i>	Arbitrary name used in conjunction with <i>username</i> and <i>acctname</i> parameters to form a fully qualified session identity. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The default is that no session name is assigned.
<i>username</i>	User name, established by the account manager, that allows you to logon under this account. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character.
<i>userpass</i>	User password, optionally assigned by the account manager. The password must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The <i>userpass</i> parameter must be preceded by a slash (/).
<i>acctname</i>	Account name as established by the system manager. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The <i>acctname</i> parameter must be preceded by a period (.).
<i>acctp</i> <i>pass</i>	Account password, optionally assigned by the system manager. The password must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The <i>acctp</i> <i>pass</i> parameter must be preceded by a slash (/).
<i>groupname</i>	Group name to be used for the local file domain and the CPU time charges as established by the account manager. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The default is your home group, if you are assigned one by the account manager. A <i>groupname</i> is required if a home group has not been assigned.
<i>grouppass</i>	Group password optionally assigned by the account manager. The password must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The <i>grouppass</i> parameter is not needed when you logon to your home group. The <i>grouppass</i> parameter must be preceded by a slash (/).

termtype or *termname* Determines terminal-type characteristics. ;TERM must be followed by a *termtype* or a *termname*. The *termtype* parameter whose assigned value determines the type of terminal used for input. MPE V/E uses this parameter to determine device-dependent characteristics such as delay factors for carriage returns. It must be a number from 0 to 22. For more information, refer to appendix A, "Terminals Supported by MPE V/E". The default value for *termtype* is assigned by the system supervisor during system configuration. This parameter is required to ensure correct listings if your terminal is not the default *termtype*.

The *termname* parameter is the name of the file containing the desired terminal-type characteristics. The file cannot have a lockword or reside on a private volume.

Users of the workstation configurator may create terminal-type files. The proper and efficient operation of a specific device by such a file is the responsibility of the user. The workstation configurator utility allows you to specify the following characteristics of your terminal: data flow control, block mode, read trigger, special characteristics, echo, line feed, parity, and printer control.

If group and/or account names are omitted, the proposed logon group and/or account name is substituted. For further information on the terminal-type characteristics, refer to the *Fundamental Communications Handbook* (5957-4634) and the *Workstation Configurator Reference Manual*(30239-90001).

cpusecs Maximum CPU time that your session can use, entered in seconds. When the limit is reached, the session is aborted. It must be a value from 1 to 32767, but cannot exceed any limit imposed by the system or account manager. To specify no limit, enter a question mark (?), UNLIM , or omit the parameter. The default is no limit.

BS,CS,DS, or ES The execution priority queue that the command interpreter uses for your session, and also the default priority for all programs executed within the session. BS is highest priority, ES is lowest, and the default is CS. If you specify a priority that exceeds the highest permitted for your account or user name by the system, MPE V/E assigns the highest priority possible below BS. DS and ES are intended primarily for batch jobs; their use for sessions is generally discouraged.

Caution



Use care in assigning the BS queue because processes in this priority class may lock out other processes. For information on the guidelines for these priority queues, refer to the TUNE command in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

HELLO

inputpriority or HIPRI Determines the input priority of the job or session. The *inputpriority* option is the relative input priority used when checking access restrictions imposed by the jobfence. The *inputpriority* option takes effect at logon time and must be from 1 (lowest priority) to 13 (highest priority). If you supply a value less than or equal to the current jobfence set by the system operator, your session is denied access. The default is 8.

The HIPRI option is used for two different purposes when logging on. You can use it to override the system jobfence or to override the session limit.

When you use the HIPRI option to override the jobfence, the system first checks to see if you have system manager (SM) or system operator (OP) capability. If you have either of these capabilities, you are logged on and your INPRI defaults to the system's jobfence and execution limit. If you do not have either of these capabilities, the system attempts to log you on using INPRI=13 and will succeed if the jobfence is 12 or less, and if the session limit is not exceeded. If you want to override the session limit (that is, to logon after the maximum number of sessions set by the operator has been reached), you can specify HIPRI, but to do so you must have either SM or OP capability. The system does not override the session limit automatically. If you use the HIPRI option without SM or OP capability, the following warning is displayed:

```
MUST HAVE 'SM' OR 'OP' CAP. TO SPECIFY HIPRI, MAXIMUM INPRI OF 13 IS USED
(CIWARN 1460)
```

Use

You may issue this command from a session. You may not use it from a job, program, or in BREAK. Pressing **BREAK** during the output of the logon message terminates the rest of the message. You remain logged on and MPE V/E prompts you for your next command, provided that you do not have an OPTION LOGON UDC. If there is an OPTION LOGON UDC that is not a program, it is forced at logon time since **BREAK** has no effect on it.

Operation

The HELLO command initiates an interactive session and must be entered from a terminal; no other device can be used for this command. You must supply both a valid user name and account name in your logon command. Otherwise, MPE V/E rejects your logon attempt and prints an error message to that effect. If your logon is accepted, however, MPE V/E confirms this by printing specific logon information and prompting you for your next MPE V/E command. This information appears in the following example, where a user has logged on under the user name MAC and the account name TECHPUBS:

```
:HELLO MAC.TECHPUBS
```

```
HP3000  RELEASE:  G.03.08   USER VERSION: G.03.08   TUE, AUG 22, 1989,  1:28 PM
MPE V    HP32033  G.03.08  Copyright Hewlett-Packard 1984.  All Rights Reserved.
```

```
:                               ** MPE V/E prompts for next command. **
```

Sometimes a welcome message from the system operator appears after MPE V/E has verified your logon.

MPE V/E assigns you a session number that uniquely identifies your session to MPE V/E and to other users. MPE V/E assigns such numbers to sessions in sequential order as they are logged on. If you are on a modem and do not logon within the time limit permitted by the system, the line is dropped. You must redial and press **(Return)** again. If you are already logged on and you issue the **HELLO** command, you are logged off your current session and onto a new session, except in the following cases:

- If spaces precede the command.
- If the command is executed using a **REDO** command.
- If the command is issued in a user-defined command (UDC).

In certain instances, you may be required to furnish information in addition to the user and account names in your **HELLO** command or () **COMMAND LOGON**. This information includes:

- Your group name.
- One or more passwords.
- Terminal-type code.

Group Name

The group you select at logon for your local file domain is known as your logon group. If your account manager has associated a home group with your user name, and if you want this group as a logon group, you need not specify it. MPE V/E automatically assigns the home group as your logon group when you logon. But if you want to use some other group as your logon group, you must specify that group's name in your logon command in this way:

:HELLO MAC.TECHPUBS,YGROUP

If your user name is not related to a home group, you must enter a group name in your **HELLO** command or () **COMMAND LOGON**, or your logon attempt is rejected.

Once you logon, if the normal (default) file security provisions of MPE V/E are in force, you have unlimited access to all files in your logon and home groups. Furthermore, you can read files and execute programs stored in the **PUB** (public) group of your account and the **PUB** group of the **SYS** (system) account. You cannot, however, access any other files in any way. For more information about files and file security, refer to the *MPE File System Reference Manual* (30000-90236).

Passwords

To enhance the security of an account, and to prevent unauthorized accumulation of charges against the account, the system manager may assign a password. Similarly, an account manager may associate passwords with the users and groups belonging to his account.

HELLO

Sometimes it is convenient to have MPE V/E prompt you for required passwords when you are logging onto the system. You do this by omitting the passwords from the logon command. When you logon, the command is printed in the usual way; MPE V/E prompts you for the password, then turns the echo off so that the password is not printed. If you enter the password incorrectly, the prompt reappears and you have two more chances to enter the password correctly. After the third incorrect entry, the line is dropped, and you must press **Return** to receive another prompt. The echo is turned on after all passwords are read.

If you are using an account, user name, or group (other than your home group) that has a password, you must furnish that password when you logon. Include the password after the name of the protected entity, separated from that name by a slash mark (/). In MPE V/E, slash marks denote passwords.

For instance, if the group XGROUP requires a password and if you wish this group as your logon group, you could enter the password in this fashion:

```
:HELLO MAC.TECHPUBS,XGROUP/XPASS
```

When you specify your home group as your logon group, you need not enter a password, even if that group has such a password.

If you have the HP Security Monitor installed, your system manager may require password prompts for your system. MPE V/E does not accept embedded passwords (as in **HELLO MAC.TECHPUBS,XGROUP/XPASS**). If you enter an embedded password, MPE V/E displays an error statement:

```
PASSWORD PROMPTS ARE REQUIRED, EMBEDDED PASSWORDS ARE NOT ALLOWED.  
(CIERR 1449)
```

Terminal Types

MPE V/E must be able to determine certain characteristics about your terminal, such as input and output speed, in order to conduct a session. If you logon using a different type of terminal than the type the system manager has configured, you must specify your terminal type when you logon. For example, if you are logging on at an HP 2621A, but the default type at your site is a HP 2645A, you would enter:

```
:HELLO MAC.TECHPUBS;TERM=10
```

Terminal types may also be specified via a file containing terminal-type characteristics. Users of the workstation configurator may create terminal type files without any restrictions.

Note



The proper and efficient operation of a specific device by a user-created terminal type is the responsibility of the user.

HELLO

For example, a terminal type that specifies no flow control mechanism generally creates data overruns on the device. Users may control each of the following terminal-type characteristics: echo, line feed during input, form feed during output, backspace response, printer control, and parity. Refer to the *Workstation Configurator Reference Manual* (30239-90001) and the *Fundamental Communications Handbook* (5957-4634) for more information. Refer also to appendix A, "Terminals Supported by MPE V/E".

Example

To start a session named ALPHA with the user MAC, the account TECHPUBS, the group XGROUP, and the group password XPASS, enter:

```
:HELLO ALPHA,MAC.TECHPUBS,XGROUP/XPASS
```

```
HP3000  RELEASE:  G.03.08   USER VERSION: G.03.08   TUE, AUG 22, 1989,  1:28 PM  
MPE V   HP32033  G.03.08   Copyright Hewlett-Packard 1984.  All Rights Reserved.
```

```
:                               ** MPE V/E then prompts for next command. **
```

HELP

Accesses the HELP subsystem.

Syntax

```
HELP [ HELP
      tablecontents
      command[ [ , ] keyword ]
      ALL
      EXIT ]
```

Parameters

HELP Displays information for the **HELP** command. This is treated the same as entering **HELP** with any other command, because **HELP** is a command.

tablecontents Any of several items for which information may be obtained. These items are:

SESSIONS
JOBS
PROGRAMS
FILES
MANAGE
OPERATOR
SPOOLER
UTILITY

If you want information on running sessions, you would enter:

:HELP SESSIONS

command Any MPE V/E command. MPE V/E displays the command name and syntax. In addition, a list of keywords for that command is displayed.

Any UDC can also be the *command*. In this case, MPE V/E lists out the lines of the UDC from the UDC catalog file and references the UDC as a “user-defined command”.

keyword One of the keywords described under the command parameter. All commands have the following keywords:

PARMS	Lists all parameters of the specified command.
OPERATION	Describes the use of the specified command.
EXAMPLE	Displays an example showing how to use the specified command.
ALL	Displays all parameters, describes the operation, and shows an example of the command.

The **FILE** command has the following additional keywords:

NAME
DEVICE
ACCESS
DISP

When you enter the **HELP** command followed by a command name and the **ALL** parameter, MPE V/E displays all information for that command (syntax, parameters, operation, and example). You can enter the keyword parameter with a command, as in **HELP HELLO PARS**. In this case, **PARS** is a keyword and must be separated from the command with a space or comma. You can also enter the keyword parameter alone if you are running the **HELP** subsystem in subsystem mode, in response to the ">" prompt.

ALL Displays the entire table of contents and the contents of each keyword for the **HELP** command.

EXIT Exits the **HELP** subsystem.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** aborts execution.

Operation

The **HELP** command accesses the **HELP** subsystem. If you enter it with no parameters, as in:

:HELP

HELP enters the subsystem mode, displays a table of contents and "greater-than" (>) prompt, and awaits your input. When you enter an item from the table of contents, such as **SESSIONS**, it produces a listing of all the commands used in running sessions. When you enter a command name, it produces the syntax for that command and a list of the keywords. When you enter a keyword such as **PARS**, it produces a listing of all the items for that keyword (all parameters, in this case).

You enter **E** or **EXIT**, or press **BREAK** to terminate the **HELP** subsystem.

Enter **CTRL Y** to terminate the display, and return to a system prompt (:). Entering **CTRL S** temporarily stops the display, and entering **CTRL Q** resumes the display again. This is useful if the screen is full, as when you use a CRT terminal. If you press **Return** in subsystem mode, MPE V/E displays information up to the next keyword or command. For example, after you enter **STORE** with the **HELP** command, MPE V/E displays **STORE** syntax, the keyword list (**PARS**, **OPERATION**, and **EXAMPLE**) and provides a prompt (>). When you press **Return** again, MPE V/E displays all parameter information for the **STORE** command. Pressing **Return** again at the next prompt causes MPE V/E to display all the information on the operation of the **STORE** command. (This is similar to turning the pages of a manual.)

HELP

If you enter HELP with a parameter, the HELP subsystem enters the “immediate” mode. Information pertaining to that parameter is displayed immediately. For example, if you enter:

:HELP ABORT

the HELP subsystem displays:

ABORT

Aborts current program or operation.

SYNTAX

:ABORT

KEYWORDS: PARMS,OPERATION,EXAMPLE

:

Example

To obtain information concerning MPE V/E utility functions, enter:

:HELP UTILITY

MPE V/E displays:

Utility Functions. Following are the commands used:

CONSOLE	SHOWALLOW
PTAPE	SHOWCATALOG
SETCATALOG	SHOWME
SETMSG	SHOWTIME
SPEED	TELL
RECALL	TELLOP
REDO	

You can use any command name as a keyword.

KEYWORDS: SESSIONS,JOBS,PROGRAMS,FILES,MANAGE,OPERATOR,SPOOLER,
UTILITY

:

IF

Used to control the execution sequence of a job.

Syntax

```
IF [ ( logexpr ) ] THEN
```

Parameters

logexpr Logical expression, consisting of the operands and relational operators. The permitted relational operators are:

- > greater than
- < less than
- >= greater than or equal to
- <= less than or equal to
- = equal to
- <> not equal to

The allowed operands are:

- JCW or CIERROR.
- Any JCW values saved in a name with the SETJCW command.
- OK, WARN, FATAL or SYSTEM, as defined under the SETJCW command.
- Integer constants.

You can form compound logical expressions using the AND and OR logical operators, and nest them within parentheses.

Use

You may issue this command from a session, job, or in BREAK. You may not use it from a program. Pressing **BREAK** has no effect on this command.

Operation

The IF command begins an IF block consisting of all the commands after the IF command up to, but not including, the next ELSE or ENDIF statement. The ELSE or ENDIF statements must be at the same nesting level as the IF statement. Another similar block can follow the ELSE statement. Nesting of the blocks is allowed to 15 levels. The ENDIF statement ends the IF block.

The logical expression is evaluated and, if it is valued as true, the IF block is executed; if false, the ELSE block (if one exists) is executed.

IF

Example

The following job listing illustrates the use of an IF statement with ELSE and ENDIF statements:

```
!CONTINUE
!SPL MYPROG,MYUSL
!IF JCW>=FATAL THEN
!  TELL MAC.TECHPUBS;COMPILE FAILED
!ELSE
!  TELL MAC.TECHPUBS;COMPILE COMPLETED
!ENDIF
```

JOB

Identifies a job to be executed in conjunction with the **STREAM** command in batch mode.

Syntax

```
JOB [ jobname, ] username [ /userpass ] . acctname
[ /acctppass ] [ , groupname [ /grouppass ] ] [ ;TIME=cpusecs ]
[ ;PRI= $\left\{ \begin{array}{l} \text{BS} \\ \text{CS} \\ \text{DS} \\ \text{ES} \end{array} \right\} \right]$  [  $\left\{ \begin{array}{l} ;\text{INPRI}=\textit{inputpriority} \\ ;\text{HIPRI} \end{array} \right\} \right]$  [ ;RESTART ]
[ ;OUTCLASS=[ device ] [ , outputpriority [ , numcopies ] ] ] [ ;TERM= $\left\{ \begin{array}{l} \textit{termtype} \\ \textit{termname} \end{array} \right\} \right]$ 
```



Parameters

<i>jobname</i>	Arbitrary name used with <i>username</i> and <i>acctname</i> parameters to form a job identity. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The default is that no job name is assigned.
<i>username</i>	User name, established by the account manager, that allows you to logon under this account. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character.
<i>userpass</i>	User password, optionally assigned by the account manager. The password must contain from one to eight alphanumeric characters, beginning with an alphabetic character.
<i>acctname</i>	Account name as established by the system manager. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The <i>acctname</i> parameter must be preceded by a period (.).
<i>acctp</i> <i>pass</i>	Account password, optionally assigned by the system manager. The password must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The <i>acctp</i> <i>pass</i> parameter must be preceded by a slash (/).
<i>groupname</i>	Group name to be used for the local file domain and for CPU time charges, as established by the account manager. The name must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The default is the home group, if one is assigned. The parameter is required if a home group is not assigned.
<i>grouppass</i>	Group password, optionally assigned by the account manager. The password must contain from one to eight alphanumeric characters, beginning with an alphabetic character. The <i>grouppass</i> parameter is not needed when you logon to your home group. The <i>grouppass</i> parameter must be preceded by a slash (/).

JOB

<i>cpusecs</i>	Maximum CPU time allowed job, in seconds. When this limit is reached, the job is aborted. It must be a value from 1 to 32767. To specify no limit, enter a question mark or UNLIM , or omit this parameter. The default is the system-configured job limit.
BS,CS,.DS, or ES	The execution priority queue that the command interpreter uses for your session. This is also the default priority for all programs executed within the session. BS is highest priority, ES is lowest, and the default is CS. If you specify a priority that exceeds the highest permitted for your account or user name by the system, MPE V/E assigns the highest priority possible below BS. DS and ES are intended primarily for batch jobs; their use for sessions is generally discouraged.

Caution



Use care in assigning the BS queue. Processes in this priority class can lock out other processes.

	For information on the guidelines for these priority queues, refer to the TUNE command in the <i>MPE V System Operation and Resource Management Reference Manual</i> (32033-90005).
<i>inputpriority</i> or HIPRI	Determines the input priority of the job or session. The <i>inputpriority</i> option is the relative input priority used in checking against access restrictions imposed by the jobfence. The <i>inputpriority</i> option takes effect at logon time and must be from 1 (lowest priority) to 13 (highest priority). If you supply a value less than or equal to the current jobfence set by the system operator, the session is denied access. The default is 8. HIPRI is the request for maximum input priority, causing the job to be scheduled regardless of the current jobfence or execution limit for jobs. This parameter overrides the <i>inputpriority</i>. You can specify this parameter only if you have system manager or system supervisor capability. If not, the system tries to log you on with INPRI=13.
RESTART	Request to restart a spooled job that has been interrupted by system termination or restart. This parameter takes effect automatically when the system is subsequently restarted with the WARMSTART option. This parameter applies only to jobs initiated on spooled input devices. It is ignored for other jobs. The default is that spooled jobs are not restarted after system termination or restart.
<i>device</i>	Class name or logical device number (<i>ldev</i>) of the device that will receive listing output. You cannot specify a magnetic tape unit. If the parameter is not a valid <i>ldev</i> or class name, an error is generated. The default is defined in the system configuration.

Note

“Non-sharable device” (ND) file access capability is required in order to use this parameter.

-
- outputpriority* The output priority for the job list file, if it is destined for a spooled line printer or card punch. This parameter is used to select the next spooled device file (on disk) for output, from those contending for a specific printer or punch. It must be a value from 1 (lowest priority) to 13 (highest priority). When the *outputpriority* is 1, output is deferred. If you want your output printed or punched from disk, you should use an *outputpriority* of 2 or greater.
- This parameter applies only to output destined for spooled output devices, and is ignored for other output. The default is 8.
- numcopies* Number of copies of the job listing to be produced. This parameter applies only when the listing is directed to a spooled device; otherwise it is ignored. If the number of copies is less than 1, a warning is issued, and the command executes with the default value of 1. If the number of copies is greater than 127, an error message is printed and 127 copies only are printed. The default is 1.
- termtype* or *termname* Determines the terminal-type characteristics. The *termtype* parameter determines the type of terminal used for input. MPE V/E uses this parameter to determine device-dependent characteristics such as delay factors for carriage returns. It must be a number from 0 to 22. Refer to appendix A, “Terminals Supported by MPE V” for a list of terminal-type numbers. The system supervisor assigns the default value for *termtype* during system configuration. This parameter is required, to ensure correct listings if your terminal is not the default *termtype*.
- The *termname* parameter is the name of the file containing the desired terminal-type characteristics. The file may not have a lockword or reside on a private volume.
- Users of the workstation configurator are allowed to create terminal-type files.

Note

The proper and efficient operation of a specific device by a user-created terminal-type file is the responsibility of the user who created the file.

The workstation configurator utility allows you to specify the following characteristics of your terminal: data flow control, block mode, read trigger, special characteristics, echo, line feed, parity, and printer control.

If group or account names are omitted, MPE V/E substitutes the proposed logon group and/or account name. For further information on the terminal-type characteristics, refer to the *Fundamental Communications Handbook* (5957-4634) and the *Workstation Configurator Reference Manual* (30239-90001).

JOB

Use

You may issue this command from a session or job. You may use it from a program or in BREAK. Pressing **BREAK** has no effect on this command.

Operation

The JOB command initiates a batch job by establishing contact with MPE V/E. If you initiate the job through a session and it is in the WAIT state, your terminal is locked until the job enters the EXEC state. When MPE V/E begins the job, it displays the following information on the list device:

- Job number, as assigned by MPE V/E, to identify the job.
- Date and time.
- "HP 3000", and both the base and modified MPE V/E *version.update.fix* numbers.

In the JOB command, as in the HELLO command, you must always include your user name and account name, which you obtain from your account manager. If you omit either of these names, or enter them incorrectly, MPE V/E rejects your job and prints an error message on the standard listing device. If your job is accepted, MPE V/E begins processing the job. If the job is entered through a spooled input device, it is copied to a disk file and initiated from that file rather than the originating device.

If the standard listing file is a line printer, MPE V/E prints a header page prior to listing the JOB command. (The system operator can disable the printing of this page via the console command HEADOFF.)

The job number assigned by MPE V/E always uniquely identifies your job to MPE V/E and other users. MPE V/E assigns such numbers in sequential order as jobs are accepted.

Sometimes, the job acceptance information includes a message from the system operator following the standard display. When present, this is the same message output as is displayed in the logon information for sessions.

The minimum information needed for job initiation is the user and account name. Under certain circumstances, however, you may need to furnish the following additional information in your JOB command:

- File group name.
- User, account, and/or group passwords.

The cases in which this information is required, and the rules for supplying it, are the same as those for the HELLO command for sessions, except that:

- When you omit a required password from the JOB command, MPE V/E rejects your access attempt without prompting you for the password.
- When you enter the JOB command through a device other than a terminal and the standard input device is different from the standard listing device, MPE V/E does not echo passwords.
- When the standard listing device is a line printer, and you do not specify a file group name, CPU time limit, execution priority, and/or input priority in the JOB command, the default values that MPE V/E assigns to the omitted parameters appear on the job listing.

A job will execute user-defined commands (UDCs) whether the job is interactive or streamed. UDCs will also execute in MPE V/E itself or in any subsystem. You must anticipate actions, prompts, or questions incorporated in any UDC you use, whether you use it interactively or in a streamed job.

The EOJ command is used to end a job. The only alternative exits from a job are an error flush or an ABORT. An error flush stops the completion of the job at the point the error occurred.

Example

To start a job named BETA with the *username* MAC, *acctname* TECHPUBS, *groupname* XGROUP, and *grouppass* XPASS, enter:

```
:JOB BETA,MAC.TECHPUBS,XGROUP/XPASS
```

```
JOB NUMBER = #J1
```

```
HP3000 / MPE V G.03.08 (BASE G.03.08) WED, JUL 26, 1989, 8:22 AM
```

```
:
```

Additional Discussion

MPE V File System Reference Manual (30000-90236)

MPE V Intrinsic Reference Manual (32033-90007)

JOBFENCE

Defines the minimum input priority that a job or session must have in order to execute.

Syntax

JOBFENCE *priorityfence*

Parameters

priorityfence A number between 0 and 14 inclusive. Within this range, smaller numbers are less limiting, larger numbers more limiting.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may issue it only from the console *unless* it is distributed to users with the **ALLOW** command.

Operation

MPE V/E does not dispatch jobs or sessions with an input priority less than or equal to the *priorityfence* until their input priority is raised with the **ALTJOB** command, or until the jobfence is lowered. System managers and system supervisors may override the jobfence setting by logging on with the **;HIPRI** parameter of the **JOB** or **HELLO** commands. They may also logon with an input priority greater than the **JOBFENCE** as reported by the **SHOWJOB** command.

Examples

To defer all non-HIPRI jobs and sessions, first set the jobfence to 14, as shown below:

```
:JOBFENCE 14
16:18/#J7/34/DEFERRED JOB INTRODUCED ON LDEV #10
16:18/#J8/35/DEFERRED JOB INTRODUCED ON LDEV #10
```

Then enter the SHOWJOB command to display the effect of the JOBFENCE command:

:SHOWJOB

JOBNUM	STATE	IPRI	JIN	JLIST	INTRODUCED	JOB NAME
#S26	EXEC		20	20	THU 4:17P	OPERATOR.SYS
#J7	WAIT	D 8	10S	12	THU 4:18P	JOB1, FIELD.SUPPORT
#J8	WAIT	D 8	10S	12	THU 4:18P	JOB2, FIELD.SUPPORT

3 JOBS:

0 INTRO

2 WAIT; INCL 2 DEFERRED

1 EXEC; INCL 1 SESSIONS

0 SUSP

JOBFENCE= 14; JLIMIT= 5; SLIMIT=16

Finally, reset the jobfence to 6, which allows waiting jobs to logon.

:JOBFENCE 6

16:21/#J7/34/LOGON FOR: JOB1, FIELD.SUPPORT ON LDEV #10

16:21/#J8/35/LOGON FOR: JOB2, FIELD.SUPPORT ON LDEV #10

JOBPRI

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

Syntax

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

Parameters

- | | |
|------------------------|--|
| <i>maxsubqueue</i> | The maximum priority at which batch jobs are allowed to run. This overrides any job priority you may have requested with the JOB command. This parameter may be ES, DS, CS, or 0 . If 0 is specified, no limit is imposed on batch jobs. The default is no change in maximum priority. |
| <i>defaultsubqueue</i> | The default execution priority for batch jobs, which may be ES, DS, or CS. This takes effect if you do not specify an execution priority in the JOB command. The default is no change in execution priority. |

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** has no effect on this command. You must have OP capability to execute this command.

Operation

The JOBPRI command lets you alter the default priority for batch jobs and set maximum limits for their execution. The *maxsubqueue* parameter specified in the JOBPRI command takes precedence over *defaultsubqueue*. Therefore, selecting a default parameter greater than the value of *maxsubqueue* parameter does not affect job execution: jobs are still initiated with the maximum priority parameter.

Example

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

```
:JOBPRI 0
```

JOBSECURITY

Designates what level of user may request resources and control the execution of jobs.

Syntax

$$\text{JOBSECURITY} \left\{ \begin{array}{l} \text{HIGH} \\ \text{LOW} \end{array} \right\}$$

Parameters

- | | |
|------|---|
| HIGH | Permits only the operator logged on at the console to use job control commands. |
| LOW | Allows individual users to exercise control over their own jobs. |

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only from the console *unless* it is distributed to users with the **ALLOW** command.

Operation

The **JOBSECURITY** command controls the use of the **ABORTJOB**, **ALTJOB**, **BREAKJOB**, and **RESUMEJOB** commands. The **HIGH** parameter permits only the operator to issue these commands. The **JOBSECURITY LOW** parameter allows any user to issue these commands for their own jobs; the job's user name and account must match that of the user. Account managers do not need a matching user name, so they may control the execution of any job in their account.

Example

To allow any user to abort, alter, **BREAK** or resume their own jobs, enter:

:JOBSECURITY LOW

LDISMOUNT

Performs the logical dismount of a private volume set or class.

Syntax

LDISMOUNT *vsname.group.account*

Parameters

<i>vsname</i>	Volume set name for which you are requesting a logical dismount.
<i>group.account</i>	The group and account under which the volume set was created.

Use

You may issue this command from a session, job, program or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only from the system console *unless* it is distributed to users with the **ALLOW** command.

Operation

The **LDISMOUNT** command is used to dismount non-system domain disks while the disk system is online. It informs MPE V/E that the volume class or set is no longer needed and that member volumes should be logically dismounted, as a unit, from the drive(s) on which they reside. The dismount request remains pending until any files being accessed on the disk volume are closed, then the command executes. If the specified volume class or set is not mounted when you attempt to logically dismount it, you receive an error message.

Example

To dismount a private volume set named **DATABASE**, which was created in the **PAYROLL** group of the **ACCTNG** account, enter:

:LDISMOUNT DATABASE.PAYROLL.ACCTNG

Additional Discussion

Refer to the discussion of "Private Volumes" in the *MPE V System Operation and Resource Management Reference Manual*(32033-90005).

LIMIT

Limits the number of concurrently running jobs and/or sessions.

Syntax

$$\text{LIMIT} \left\{ \begin{array}{l} \text{numberjobs} \\ , \text{numbersessions} \\ \text{numberjobs}, \text{numbersessions} \end{array} \right\}$$

Parameters

<i>numberjobs</i>	The number of jobs.
<i>numbersessions</i>	The number of sessions.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may issue it only from the system console *unless* it is distributed to users with the **ALLOW** command.

Operation

The system supervisor establishes maximum job and session limits during system configuration. Within these boundaries, the system operator may use the **LIMIT** command to redefine the job and session limits. When the system is restarted from disk in a **WARMSTART**, these limits are retained. When any other startup option is used, the values configured by the supervisor take effect.

You must include at least one parameter in the **LIMIT** command. If you execute the **LIMIT** command without parameters, the following message is displayed on the console:

EXPECTED ONE OF JOB OR SESSION LIMIT. (CIERR 3071)

If you enter one parameter and omit the other, the limit of the omitted parameter remains unchanged.

No new jobs or sessions are dispatched that would cause either of these limits to be exceeded, unless they are initiated with the **;HIPRI** parameter of the **JOB** or **HELLO** commands. Jobs that are not 'HIPRI' can be introduced once the limit is achieved, but they do not execute.

If you attempt to logon to a non-HIPRI session after the limit has been reached, you receive the message:

CAN'T INITIATE NEW SESSIONS NOW

The specified limits may be exceeded at the time the command is issued. This does not cause jobs or sessions executing at the time to abort. They continue to execute, but no new jobs are allowed to enter the executing state, and no new sessions are initiated.

LIMIT

Examples

To limit the number of jobs to 2 and the number of sessions to 15, enter:

```
:LIMIT 2,15  
:SHOWJOB
```

JOBNUM	STATE	IPRI	JIN	JLIST	INTRODUCED	JOB NAME
#S24	EXEC		20	20	TUE 1:54A	OPERATOR.SYS
#S26	EXEC		177	177	TUE 5:01A	CHEWY,RSPOOL.SYS
#S96	EXEC	QUIET	35	35	TUE 8:31A	SLIDES.SIMON

```
3 JOBS:  
  0 INTRO  
  0 WAIT; INCL 0 DEFERRED  
  3 EXEC; INCL 3 SESSIONS  
  0 SUSP  
JOBFENCE= 6; JLIMIT= 2; SLIMIT= 15
```

To limit the number of sessions to 13, but retain the current job limit, enter:

```
:LIMIT ,13
```

LISTACCT

Lists the attributes for one account, with AM (account manager) capability, or for any or all accounts, with SM (system manager) capability.

Syntax

```
LISTACCT [ { @
           acct } ] [ , listfile ]
```

Parameters

@	Specifies all accounts. This is the default for users with SM capability, which is required to list all accounts.																				
<i>acct</i>	The name of the account whose attributes are to be listed. Account managers may specify only their own account. The default is the logon account. The characters “@”, “#”, and “?” can be used as wildcard characters, but count toward the eight-character limit. These wildcard characters have the following meanings: <table> <tr> <td>@</td><td>Specifies zero or more alphanumeric characters and denotes all members of the set.</td></tr> <tr> <td>#</td><td>Specifies one numeric character.</td></tr> <tr> <td>?</td><td>Specifies one alphanumeric character.</td></tr> </table> You can use the characters as follows: <table> <tr> <td>n@</td><td>List all accounts starting with the character “n”.</td></tr> <tr> <td>@n</td><td>Lists all accounts ending with the character “n”.</td></tr> <tr> <td>n@x</td><td>Lists all accounts starting with the character “n” and ending with the character “x”.</td></tr> <tr> <td>n## ... #</td><td>Lists all accounts starting with the character “n” followed by up to seven digits.</td></tr> <tr> <td>?n@</td><td>Lists all accounts whose second character is “n”.</td></tr> <tr> <td>n?</td><td>List all two-character accounts starting with the character “n”.</td></tr> <tr> <td>?n</td><td>List all two-character accounts ending with the character “n”.</td></tr> </table>	@	Specifies zero or more alphanumeric characters and denotes all members of the set.	#	Specifies one numeric character.	?	Specifies one alphanumeric character.	n@	List all accounts starting with the character “n”.	@n	Lists all accounts ending with the character “n”.	n@x	Lists all accounts starting with the character “n” and ending with the character “x”.	n## ... #	Lists all accounts starting with the character “n” followed by up to seven digits.	?n@	Lists all accounts whose second character is “n”.	n?	List all two-character accounts starting with the character “n”.	?n	List all two-character accounts ending with the character “n”.
@	Specifies zero or more alphanumeric characters and denotes all members of the set.																				
#	Specifies one numeric character.																				
?	Specifies one alphanumeric character.																				
n@	List all accounts starting with the character “n”.																				
@n	Lists all accounts ending with the character “n”.																				
n@x	Lists all accounts starting with the character “n” and ending with the character “x”.																				
n## ... #	Lists all accounts starting with the character “n” followed by up to seven digits.																				
?n@	Lists all accounts whose second character is “n”.																				
n?	List all two-character accounts starting with the character “n”.																				
?n	List all two-character accounts ending with the character “n”.																				
<i>listfile</i>	Destination of the attribute listing. The default is \$STDLIST, but the output may be redirected with the FILE command.																				

LISTACCT

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have system manager (SM) or account manager (AM) capability to execute this command.

Operation

The **LISTACCT** command lists the attributes for one account (AM capability) or for any or all accounts (SM capability).

The listing appears as an octal dump of the account entry; each is headed by **A = " "** and the account name. When the dump contains alphanumeric data, a translation into ASCII appears to the right of each line. The following is a sample listing of an octal dump:

```
A = SYS
051531 051440 020040 020040 000006 000007 177607 000713 SYS.....
000000 000000 020040 020040 020040 020040 000010 052150 .....Th
077777 177777 000010 113257 077777 177777 000037 147337 .....
077777 177777 004551 140036 000123 000217 .....i...S..
```

Each octal word in the preceding listing is displayed in the following decimal representation of the dump. The following table decodes the information.

```
--00-- --01-- --02-- --03-- --04-- --05-- --06-- --07--
--08-- --09-- --10-- --11-- --12-- --13-- --14-- --15--
--16-- --17-- --18-- --19-- --20-- --21-- --22-- --23--
--24-- --25-- --26-- --27-- --28-- --29--
```

Words	Content
0-3	Account name.
4-5	Account's group and user index pointers.
6-7	Account attributes.
8-9	Local attributes.
10-13	Password-encrypted.
14-15	Permanent file space usage count (in sectors).
16-17	Permanent file space limit (in sectors).
18-19	CPU time usage count (in seconds).
20-21	CPU 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).
28	Command file location of account UDCs.
29	Command file location of SYS account UDCs.

Example

To list the attributes of the account SYS ,enter:

:LISTACCT SYS

A = SYS

```
051531 051440 020040 020040 000006 000007 177607 000713 SYS.....
000000 000000 020040 020040 020040 020040 000010 052150 .....Th
077777 177777 000010 113257 077777 177777 000037 147337 .....
077777 177777 004551 140036 000123 000217 .....i...S..
:
```

LISTEQ

Displays all active file equations for a job or session.

Syntax

```
LISTEQ [ listfile ]
```

Parameters

listfile The file to which output is directed. The default is \$STDLIST.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** aborts execution.

Operation

The LISTEQ command displays all the active file equations for a job or session.

Example

An example of LISTEQ is given below.

```
:LISTEQ
```

```
*** FILE EQUATIONS
```

```
FILE TAPE1;DEV=ATAPE
```

```
FILE PP;ENV=LP2.ENV.OSE;DEV=EPOC
```

LISTF

Lists descriptions of one or more permanent disk files.

Syntax

$$\text{LISTF } [\text{fileset}] \left[\begin{matrix} \left(\begin{matrix} , 0 \\ , 1 \\ , 2 \end{matrix} \right) \\ \left(\begin{matrix} , -1 \\ , -2 \end{matrix} \right) \end{matrix} \right] [; \text{listfile}]$$

Parameters

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

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

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

- @ Specifies one or more alphanumeric characters. When used by itself, “@” denotes all members of the set.
- # Specifies one numeric character.
- ? Specifies one alphanumeric character.

These characters can be used with files as follows:

- n@ Lists all files starting with the character “n”.
- @n Lists all files ending with the character “n”.
- n@x Lists all files starting with the character “n” and ending with the character “x”.
- n##### Lists all files starting with the character “n” followed by seven digits.
- ?n@ Lists all files whose second character is “n”.
- n? Represents all two-character files starting with the character “n”.
- ?n Represents all two-character files ending with the character “n”.

Depending on the position of these characters in the *fileset* parameter, they can also be used to indicate groups and accounts. If you do not specify a group or account, MPE V/E assumes the logon group and account. The default is “@” (lists all files in logon group).

LISTF

0, 1, 2, -1, or -2	Displays information about the file. Displays only the file name. The default is 0 if no list level is specified.
0	Displays only the file name. The default is 0 if no list level is specified.
1	Displays the file name, file code, record size, format, the current end-of-file location, and the maximum number of records allowed in the file. It also shows whether the file is ASCII or binary; CCTL or NOCCTL; CIR, MSG, or STD. An asterisk following the file name indicates that someone has performed an FOPEN on it and not yet done an FCLOSE.
2	Displays all the information found with option 1, plus the blocking factor, the number of disk sectors in use (including those in use for system and user labels), the number of extents currently allocated, and the maximum number of extents allowed.
-1	Displays the octal listing of the file label. The first line of this listing is the directory entry for the file being listed. The remainder is the file label. The -1 option is available only to users who have system manager (SM) or account manager (AM) capability.
-2	Displays the access control definition (ACD) information associated with the requested <i>fileset</i> . If no ACD information exists for a file or if you do not have access to the ACD (SM, AM, CR, or RACD access), a message stating that fact is issued.

listfile The name of the output file to which the descriptions are written. It is automatically specified as an ASCII file with variable-length records and these characteristics: closed in the temporary domain, CCTL, OUT access mode, and EXC (exclusive access) option. The remaining characteristics are those obtained from the FILE command default specifications. This file is temporary and cannot be overwritten by a BUILD command. The default is \$STDLIST.

Caution



When you specify levels 0, 1, 2, -1 or -2 the system directory may be locked for a brief time, so no other users can access it.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** aborts execution.

Operation

The **LISTF** command applies only to permanent disk files. It lists descriptions of one or more disk files at the level of detail you select. You need not have access to a file to list a description of it, but you must have access to it to receive information on any associated ACDs. However, a file description is not listed unless the file's home volume set (PV) is mounted. A standard user may list level 0, 1, and 2 information for any file in the system, or -2 ACD data for those files for which they have **READ ACD** access. Users with AM capability may list level -1 or -2 data for files in their own accounts. Users with SM capability may list -1 or -2 data for any file in the system.

If you send a **LISTF** command to the line printer or magnetic tape, the date and time are printed at the top of the data.

You may request that the file information be displayed on devices other than the standard listing device. Name the desired device in the **FILE** command as follows:

```
:FILE PRTR;DEV=LP      ** Equates name PRTR with device class **
                        ** name LP. **
:LISTF @.@,2;*PRTR     ** Directs level-2 description of all files **
                        ** in all groups of the logon account to PRTR. **
```

If MPE V/E fails to locate a requested file, the following error message is displayed.

```
NON-EXISTENT FILE (CIERR 907)
```

If you have a wildcard character in your request and MPE V/E cannot locate any files satisfying those conditions, the following warning is displayed:

```
NO FILES FOUND IN FILE-SET (CIWARN 431)
```

Examples

The following examples illustrate several uses of the **LISTF** command.

Level 0 (Default) Output Format

```
:LISTF
```

```
FILENAME
```

APPB	APPC	APPD	APPE	APPF	CONMSG
CONOP1	CONOP4	CONOP5	CONOPEX	CONSG	EXAMPLES
GIMAGE	GIMCOM	GIMDOC	GIMDS	GIMPO	INDEX
K10130955	LINDA	LIST	LOGON	MEMO	MTS
MTS3000	OPFRONT	PS	REPLY	SCHED	VIP

LISTF

Level 1 Output Format

:LISTF,1

ACCOUNT= LEWIS GROUP= PUB

FILENAME	CODE	-----	LOGICAL RECORD-----		
		SIZE	TYP	EOF	LIMIT
CONOP1		80B	FA	10	10
K0880909	TDPQ	104B	FA	636	2269
BOXES	DRAW	128W	FB	84	84
EXAMPLES		80B	FA	20	20
ERRORLOG		80B	FAO	12	1024

Level 2 Output Format

:LISTF,2

ACCOUNT= USERS GROUP= BARBARA

FILENAME	CODE	-----	LOGICAL RECORD-----	SPACE----				
		SIZE	TYP	EOF	LIMIT	R/B	SECTORS	#X
APPB		88B	FA	414	414	16	162	14 14
APPM		80B	VAM	119	240	16	12	8 8
APPC		80B	FAO	127	208	16	70	8 8
APPK	KSAM	80B	FA	16	200	16	20	2 7

The column headings have the following meanings:

FILENAME Gives the filename. An asterisk (*) following the filename in options 1 and 2 indicates the file is open for **READ** or **WRITE** access.

CODE Refer to the **BUILD** or **FILE** command for an explanation of the file code.

SIZE The size of the records under the heading **SIZE**, indicating the number of words (**W**) or bytes (**B**).

TYP The first letter under this heading contains information on the record format. This is listed in one of the following ways: **F** (fixed length), **V** (variable length), or **U** (undefined length).

The second letter indicates whether the file is **A** (ASCII) or **B** (binary).

The third letter indicates the kind of file. This information appears in one of the following ways: Blank (standard file unless **KSAM** appears in the code field), **O** (circular file), **M** (message file), or **R** (relative I/O file).

The letter **C** follows if the file has the ;**CCTL** attribute.

EOF	End of file location.
LIMIT	Maximum number of records allowed.
R/B	Blocking factor.
SECTORS	Sectors in use.
#X	Extents allocated.
MX	Maximum number of extents allowed.



Level -1 Output Format

:LISTF UDCS, -1

F = UDCS

```

052504 041523 020040 020040 001001 120353 UDCS.....
052504 041523 020040 020040 041517 046504 020040 020040 UDCS....COMD....
045440 020040 020040 020040 046501 051111 051523 040440 K.....TECH.
020040 020040 020040 020040 020202 004040 000001 123525 .....U
123525 123525 000000 000000 000014 000000 000000 002163 .U.U.....s
000000 000000 120173 021427 000005 177650 002374 005015 .....
000024 000036 000000 002163 001001 120353 000400 104644 .....s.....
001401 032743 001001 162503 001401 036363 001401 041324 ..5....C.....B.
001001 164214 001401 042166 001002 001740 001401 044134 .....Dv.....H.
001002 002656 001401 045323 001002 006740 000400 150014 .....J.....
000000 000000 000000 000000 000000 000000 000000 000000 .....
000000 000000 000000 000000 000000 000000 000000 000000 .....
000000 000000 000000 000000 000000 000000 000000 000000 .....
000000 000000 000000 000000 000000 000000 000000 000000 .....
000000 000000 000000 000000 004072 034411 123525 000000 .....9..U..
000000 000000 000000 000000 000000 000000 004073 002010 .....
000000 000000 000000 000000 042111 051503 020001 000001 .....DISC....

```

LISTF

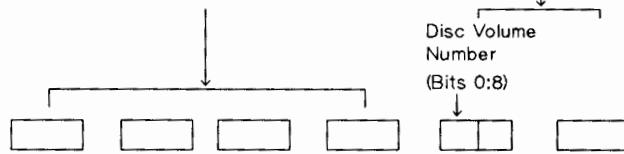
A pictorial explanation of the level -1 output, follows:

:LISTF filename, -1

F = filename

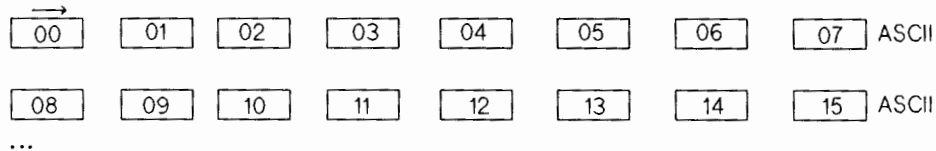
File Name (4 words-octal)

Sector address of label



Beginning of octal File Label

(Word 0)



Level -2 Output Format

:LISTF, -2

ACCOUNT= TEST GROUP= STGAMES

FILENAME -----ACD ENTRIES-----

```
APPE                      JOE.DOE : R
                         @.OSE : R,W,A,L,X
                         @.@ : X
NOACCESS                      NO ACD ACCESS
NONE                           NO ACDS
```

The disk file label contains the following:

Words		Content
0-3		Local filename
4-7		Group name
8-11		Account name
12-15		Identity for file creator
16-19		File lockword
20-21		File security matrix
22	(Bits 0:7)	Reserved for NLS
	(Bits 8:13)	Not used
	(Bits 14:1)	STORE/RESTORE release bit
	(Bits 15:1)	File secure bit: If 1, file secured If 0, file released
23		File creation date*
24		Last access date*
25		Last modification date*
26		File code
27		Private volume information
	(Bits 0:1)	Class flag bit
	(Bits 1:3)	Not used
	(Bits 4:4)	Mounted volume table index
	(Bits 8:8)	Volume mask
28	(Bit 0:1)	STORE bit (If on, STORE or RESTORE , in progress)
	(Bit 1:1)	RESTORE bit (If on, RESTORE in progress)
	(Bit 2:1)	Load bit (If on, program file is loaded)
	(Bit 3:1)	EXCLUSIVE bit (If on, file is opened with EXCLUSIVE access)
	(Bits 4:4)	Device subtype
	(Bits 8:6)	Device type
	(Bits 14:1)	File is open for WRITE
	(Bits 15:1)	File is open for READ
29	(Bits 0:8)	Number of user labels written
	(Bits 8:8)	Number of user labels
30-31		Maximum number of logical records
32-33		File control block vector
34		Checksum
35		Coldload identity
36		Foptions specifications
37		Logical record size (in negative bytes)
38		Block size (in words)

LISTF

Words		Content
39	(Bits 0:8)	Sector offset to data
	(Bits 8:3)	Not used
	(Bits 11:5)	Number of extents minus 1
40		Logical size of last extent size in sectors
41		Extent size
42-43		Number of logical records in file
44-107		Two-word addresses of up to 32 disk extents, beginning with address of first extents (words 44-45)
108-109		File allocation time
110		File allocation date
111		Not used
112-113		Start of file block number
114-115		Block number of last block
116-117		Number of open and closed records
118-119		Last modify time
124-127		Device class

*Dates are in the same format as the value returned from the **CALENDAR** intrinsic:

Bits	0	6	7	15
	Year of Century		Day of Year	

Examples

To list all of the files in your logon group that have **X** as the second alphabetic character in their names, enter:

:LISTF ?X@

FILES FOR JASON.MANU,PEN

EXAMP

:

To list all of the three-character files in your logon group and account that end in the letter **N**, enter:

:LISTF ??N

FILES FOR JASON.MANU,PEN

SYN

:

LISTFTEMP

Lists the descriptions of one or more temporary disk files for a job or session.

Syntax

```
LISTFTEMP [fileset] [ , listlevel] [ ; listfile ]
```

Parameters

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

```
filename[.groupname[.acctname]]
```

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

- @ Specifies zero or more alphanumeric characters. When used by itself, “@” denotes all members of the set.
- # Specifies one numeric character.
- ? Specifies one alphanumeric character.

These characters can be used with files as follows:

- n@ Lists all files starting with the character “n”.
- @n Lists all files ending with the character “n”.
- n@x Lists all files starting with the character “n” and ending with the character “x”.
- n##### Lists all files starting with the character “n” followed by seven digits.
- ?n@ Lists all files whose second character is “n”.
- n? Represents all two-character files that start with the character “n”.
- ?n Represents all two-character files that end with the character “n”.

Depending on their placement in the *fileset* parameter, these characters may also be used to indicate groups and accounts. If you specify nothing for the group or account, MPE V/E assumes the logon group and account. The default is “@” (all files in logon group).

LISTFTEMP

<i>listlevel</i>	Displays information about the file. The options are level 0, 1, 2, or -1:
0	Displays only the filename. The default is 0 if no listlevel is specified.
1	Displays the filename, file code, record size, format, the current end-of-file location, and the maximum number of records allowed in the file. (It also shows whether the file is ASCII or binary; CCTL or NOCCTL; CIR, MSG, or STD).
2	Displays all of the information found with the 1 option, plus the blocking factor, the number of disk sectors in use (including those in use for file labels and user headers), the number of extents currently allocated, and the maximum number of extents allowed.
-1	Displays the octal listing of the file label. The first line of this listing is the directory entry for the file being listed. The -1 option is available only to users who have SM or AM capability.
<i>listfile</i>	The name of the output file to which the descriptions are written. It is automatically specified as an ASCII file with variable-length records and these characteristics: closed in the temporary domain, CCTL (user supplied carriage-control characters), OUT access mode, and EXC (exclusive access) option. The remainder of its characteristics are those obtained from the FILE command default specifications. This file is temporary and cannot be overwritten by a BUILD command. The default is \$STDLIST.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** aborts execution.

Operation

The **LISTFTEMP** command lists descriptions of one or more temporary disk files at the level of detail you select. A file description is not listed unless the file's home volume set (PV) is mounted. A standard user may specify level 0, 1 and 2 information for any file in the system. A user with account manager or system manager capability may list level -1 data for files in his or her own account. When you send **LISTFTEMP** data to the line printer or magnetic tape, the date and time appear at the beginning of the listing.

LISTFTEMP displays the system file **\$OLDPASS** only if wildcard characters are used in the file specification and if **\$OLDPASS** fits the specification.

You may have the file information displayed on a device other than your standard list device. To do so, name the desired device in the **FILE** command, then backreference the formal file name on the **LISTFTEMP** command line. In the following example, the filename **PRTR** is equated with the device class **LP**. Then, a level 2 description of all files in all groups of the logon account is sent to **PRTR**.

```
:FILE PRTR;DEV=LP  
:LISTFTEMP 0.0,2;*PRTR
```

If MPE V/E fails to locate the file you request, one of the following error messages is displayed:

```
TEMPORARY FILE NOT FOUND (CIWARN 3289)
```

```
NO TEMP FILES EXIST (CIWARN 3284)
```

If you use a wildcard character in your request and MPE V/E cannot locate any files satisfying those conditions, the following warning is displayed:

```
TEMPORARY FILE NOT FOUND (CIWARN 3289)
```

LISTFTEMP requires 8K of stack space to execute. If you use the command programmatically, make the appropriate *mazdata* allowances for your application.

Examples

The following examples illustrate several uses of the **LISTFTEMP** command.

Level 0 (Default) Output Format

```
:LISTFTEMP 0.0.0
```

```
TEMPORARY FILES FOR JASON.MANU,PEN
```

```
$OLDPASS.PEN.MANU  
AJ.PEN.MANU  
AJJ.PEN.MANU  
Z10Z414A.JON.MANU  
TEMPFILE.TEMPGROUP.SYS.
```

LISTFTEMP

Level 1 Output Format

:LISTFTEMP BQ,1

TEMPORARY FILES FOR JASON.MANU,PEN
ACCOUNT= MANU GROUP= PEN

FILENAME	CODE	-----LOGICAL RECORD-----			
		SIZE	TYP	EOF	LIMIT
BONOP1		80B	FA	10	10
BOTE	TDPQ	104B	FA	636	2269 (TEMP)
BOXES	DRAW	128W	FB	84	84 (TEMP)
BRAMPLES*		80B	FA	20	20 (TEMP)
BRRORLOG		80B	FA0	12	1024 (TEMP)

Level 2 Output Format

:LISTFTEMP AQ,2

TEMPORARY FILES FOR JASON.MANU,PEN
ACCOUNT= MANU GROUP= PEN

FILENAME	CODE	-----LOGICAL RECORD-----				-----SPACE-----			
		SIZE	TYP	EOF	LIMIT	R/B	SECTORS	#X	MX
APPB		88B	FA	414	414	16	162	14	14 (TEMP)
APPM		80B	VAM	119	240	16	12	8	8 (TEMP)
APPC		80B	FA0	127	208	16	70	8	8 (TEMP)
APPK	KSAM	80B	FA	16	200	16	20	2	7 (TEMP)

The column headings have the following meanings:

HEADING	MEANING
FILENAME	Displays the filename. An asterisk (*) following the filename in the level 1 and 2 options indicates the file is open for READ or WRITE access.
CODE	Indicates a special file format. Refer to the BUILD or FILE command for a list of file codes and their meaning.
SIZE	Displays the size of the records under the heading SIZE , indicating the number of words (W) or bytes (B).
TYP	The first column under this heading contains information concerning record format. This is listed in one of the following ways: F (fixed length), Vn(variable length), or U (undefined length). The second column indicates whether the file is A (ASCII) or B (binary). The third column indicates the kind of file. This information appears in one of the following ways: Blank (standard file unless KSAM appears in the code field), O (circular file), D (message file) or R (relative I/O file). In the fourth column a C appears only if the ;CCTL parameter is selected.
EOF	End of file location.
LIMIT	Maximum number of records allowed.
R/B	Blocking factor.
SECTORS	Sectors in use.
#X	Extents allocated.
MX	Maximum number of extents allowed.
(TEMP)	Temporary files.

LISTFTEMP

Level -1 Output Format

:LISTFTEMP UDCS, -1

F = UDCS

(TEMPORARY FILES)

```
052504 041523 020040 020040 001001 120353 UDCS.....
052504 041523 020040 020040 041517 046504 020040 020040 UDCS....CMD....
045440 020040 020040 020040 046501 051111 051523 040440 K.....TECH.
020040 020040 020040 020040 020202 004040 000001 123525 .....U
123525 123525 000000 000000 000014 000000 000000 002163 .U.U.....s
000000 000000 120173 021427 000005 177650 002374 005015 .....
000024 000036 000000 002163 001001 120353 000400 104644 .....s.....
001401 032743 001001 162503 001401 036363 001401 041324 ..5....C.....B.
001001 164214 001401 042166 001002 001740 001401 044134 .....Dv.....H.
001002 002656 001401 045323 001002 006740 000400 150014 .....J.....
000000 000000 000000 000000 000000 000000 000000 000000 .....
000000 000000 000000 000000 000000 000000 000000 000000 .....
000000 000000 000000 000000 000000 000000 000000 000000 .....
000000 000000 000000 000000 000000 000000 000000 000000 .....
000000 000000 000000 000000 004072 034411 123525 000000 .....9..U..
000000 000000 000000 000000 000000 000000 004073 002010 .....
000000 000000 000000 000000 042111 051503 020001 000001 .....DISC....
```

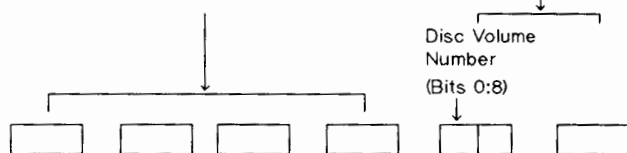
A pictorial explanation of the level -1 output, follows:

:LISTFTEMP *filename*, -1

F = filename

File Name (4 words-octal)

Sector address of label



Beginning of octal File Label

(Word 0)

```

00 01 02 03 04 05 06 07 ASCII
08 09 10 11 12 13 14 15 ASCII
...
```

The disk file label contains the following:

Words		Content
0-3		Local filename
4-7		Group name
8-11		Account name
12-15		Identity of file creator
16-19		File lockword
20-21		File security matrix
22	(Bits 0:7)	Reserved for NLS
	(Bits 8:13)	Not used
	(Bits 14:1)	STORE/RESTORE release bit
	(Bits 15:1)	File secure bit: If 1, file secured If 0, file released
23		File creation date*
24		Last access date*
25		Last modification date*
26		File code
27		Private volume information
	(Bits 0:1)	Class flag bit
	(Bits 1:3)	Not used
	(Bits 4:4)	Mounted volume table index
	(Bits 8:8)	Volume mask
28	(Bit 0:1)	STORE Bit (If on, STORE or RESTORE in progress)
	(Bit 1:1)	RESTORE Bit (If on, RESTORE in progress)
	(Bit 2:1)	Load Bit (If on, program file is loaded)
	(Bit 3:1)	EXCLUSIVE Bit (If on, file is opened with EXCLUSIVE access)
	(Bits 4:4)	Device subtype
	(Bits 8:6)	Device type
	(Bits 14:1)	File is open for WRITE
	(Bits 15:1)	File is open for READ
29	(Bits 0:8)	Number of user labels written
	(Bits 8:8)	Number of user labels
30-31		Maximum number of logical records
32-33		File control block vector
34		Checksum
35		Coldload identity
36		Foptions specifications
37		Logical record size (in negative bytes)
38		Block size (in words)

LISTFTEMP

Words		Content
39	(Bits 0:8)	Sector offset to data
	(Bits 8:3)	Not used
	(Bits 11:5)	Number of extents minus 1
40		Logical size of last extent size in sectors
41		Extent size
42-43		Number of logical records in file
44-107		Two-word addresses of up to 32 disk extents, beginning with address of first extents (words 44-45)
108-109		File allocation time
110		File allocation date
111		Not used
112-113		Start of file block number
114-115		Block number of last block
116-117		Number of open and closed records
118-119		Last modify time
124-127		Device class

* Dates are in the same format as the value returned from the **CALENDAR** intrinsic:

Bits	0	6	7	15
	Year of Century		Day of Year	

General Examples

To list all of the files in your logon group that have **X** as the second alphabetic character in their names, enter:

:LISTFTEMP ?X0

TEMPORARY FILES FOR JASON.MANU,PEN

EXAMP

:

To list all of the three-character files in your logon group and account that end in the letter **N**, enter:

:LISTFTEMP ??N

TEMPORARY FILES FOR JASON.MANU,PEN

SYN

:

LISTGROUP

Prints a list of attributes for groups.

Syntax

$$\text{LISTGROUP} \left\{ \begin{array}{l} \text{group.acct} \\ \text{group} \\ \text{@.acct} \\ \text{@} \\ \text{@.@} \end{array} \right\} [, \text{listfile}]$$

Parameters

- @** Specifies that all groups in the logon account are to be listed. This is the default.
- @.@** Specifies that all groups in all accounts are to be listed. Only users with SM capability may specify this option. The default is "@".
- group.acct** Specifies the group and account to be listed. Account managers cannot specify an account other than their own.
- group** Specifies the group (in the logon account) to be listed. The default is "@".
- @.acct** Specifies that all groups in the designated account are to be listed. If an account manager specifies an account, it must be his or her own account. The default is "@".

The characters "@", "#", and "?" can be used as wildcard characters, but count toward the eight-character limit. These wildcard characters have the following meanings:

- @** Specifies one or more alphanumeric characters or denotes all members of the set.
- #** Specifies one numeric character.
- ?** Specifies one alphanumeric character.

The characters can be used as follows:

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

LISTGROUP

listfile Destination of the attribute listing. The default is \$STDLIST, but the output may be redirected with the FILE command.

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** aborts execution. You must have account manager (AM) or system manager (SM) capability to execute this command.

Operation

The LISTGROUP command lists the attributes for all groups in one account (AM capability) or all groups in all accounts (SM capability). Also available is information on groups spanned to private volumes (PV), whether or not the volumes are mounted when the information is requested.

A sample octal dump appears below. The listing includes both group and system information, headed by "G = " and the group's name. A translation of those bytes that contain alphanumeric ASCII characters appears to the right of each line of the octal dump.

```
G = PUB
050125 041040 020040 020040 000514 020040 020040 020040 PUB.....L.....
020040 000000 001026 077777 177777 000000 046536 077777 .....M...
177777 000002 006457 077777 177777 020410 041010 000601 .....B...
000000 000516 020040 020040 020040 020040 020040 020040 ...N.....
020040 020040 020040 020040 020040 020040 000000 000000 .....
000000
```

Below is the octal listing translated into its decimal representation:

```
--00-- --01-- --02-- --03-- --04-- --05-- --06-- --07--
--08-- --09-- --10-- --11-- --12-- --13-- --14-- --15--
--16-- --17-- --18-- --19-- --20-- --21-- --22-- --23--
--24-- --25-- --26-- --27-- --28-- --29-- --30-- --31--
--32-- --33-- --34-- --35-- --36-- --37-- --38-- --39--
--40--
```

The information in the octal dump is decoded in the following table:

Words	Content
0-3	Group name, in ASCII.
4	File index pointer.
5-8	Password.
9-10	Permanent file space usage count (in sectors).
11-12	Permanent file space limit (in sectors).
13-14	CPU time usage count (in seconds).
15-16	CPU 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.
24	Group directory base linkage.
25	Group volume set definition index.
26-29	Definition's account name.
30-33	Definition's group name.
34-37	Definition's volume set name.
38	GSAVEFIPNTR
39	GMOUNTRECNR
40	GSPARE

Where:

- GSAVEFIPNTR contains the original group file index pointer or an address relative to the base of the SYSVS (system volume set) directory.
- GMOUNTRECNR is a counter which keeps track of the number of times a group's home volume set has been bound.
- GSPARE is an MPE V/E reserved word.
- The group name and password are eight-character names, right-padded with blanks.
- The double-word numeric quantities are double-word integers, with %17777777777 representing "unlimited".
- Group attributes duplicate the second word of the double-word capability returned by the -WHO intrinsic. Refer to the *MPE V Intrinsics Reference Manual* (32033-90007).

Example

To list the attributes of the PUB group in his or her own account, the account manager enters:

```
:LISTGROUP  PUB
```

```
G = PUB
050125 041040 020040 020040 000514 020040 020040 020040 PUB.....L.....
020040 000000 001026 077777 177777 000000 046536 077777 .....M...
177777 000002 006457 077777 177777 020410 041010 000601 .....B...
000000 000516 020040 020040 020040 020040 020040 020040 ...N.....
020040 020040 020040 020040 020040 020040 000000 000000 .....
000000
```

LISTLOG

Lists currently active logging identifiers on the system and whether log file changing has been enabled.

Syntax

```
LISTLOG [logid [ ;PASS ] ]
```

Parameters

- logid* The specific logging identifier to be verified. The default is to list all currently active logging identifiers on the system.
- PASS Causes the password associated with the logging identifier to be displayed. The password can be used only by the creator of the logging identifier.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have user logging (LG) capability to execute this command.

Operation

The LISTLOG command lists the logging identifier specified with its associated creator and log file. The column labeled **CHANGE** indicates whether the **CHANGELOG** command is permitted: that is, whether the name of the first logging file ends in "001" and thus follows the naming convention required by the **CHANGELOG** command. The column labeled **AUTO** indicates whether an automatic **CHANGELOG** is permitted: that is, whether the **AUTO** parameter has been specified with a **GETLOG** or **ALTLOG** command.

If the *logid* parameter is not entered, all logging identifiers on the system are displayed with their creators and log files. The **PASS** parameter, which can be used only by the creator of the logging identifier specified, causes the password associated with the logging identifier to be listed. You must have system supervisor (OP) or user logging (LG) capability in order to use this command.

Example

To list all logging identifiers on the system, enter:

```
:LISTLOG
```

LOGID	CREATOR	CHANGE	AUTO	CURRENT LOGFILE
TESTLOG	LALITHA.MPEM	YES	YES	LAL001.PEJAVAR.MPEM
TEST1	MARK.MPEM	YES	NO	M001.KSAM3000.MPEM
TEST2	PAT.MPEM	NO	NO	TEST.ALVAREZ.MPEM

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005)

MPE V Ininsics Manual (32033-90007).

Refer also to the **ALTLOG**, **CHANGELOG**, and **GETLOG** commands in this chapter.

LISTUSER

Lists attributes currently assigned to users.

Syntax

LISTUSER $\left\{ \begin{array}{l} \text{user.acct} \\ \text{user} \\ \text{@.acct} \\ \text{@} \\ \text{@.@} \end{array} \right\} [, \text{listfile}]$

Parameters

<i>user.acct</i>	The specified user in the specified account. If account managers specify <i>acct</i> , it must be their own account.																				
<i>user</i>	The specified user in the logon account.																				
<i>@.acct</i>	All users in the specified account. Account managers may specify their own account only.																				
<i>@</i>	All users in the logon account. This is the default.																				
<i>@.@</i>	All users in all accounts. (The attributes of the account are also listed.) This option can be specified only by users with system manager (SM) capability. The characters “@”, “#”, and “?” can be used as wildcard characters, but count toward the eight-character limit. These wildcard characters have the following meanings: <table><tr><td><i>@</i></td><td>Specifies zero or more alphanumeric characters and denotes all members of the set.</td></tr><tr><td><i>#</i></td><td>Specifies one numeric character.</td></tr><tr><td><i>?</i></td><td>Specifies one alphanumeric character.</td></tr></table> The characters can be used as follows: <table><tr><td><i>n@</i></td><td>List all users starting with the character “n”.</td></tr><tr><td><i>@n</i></td><td>List all users ending with the character “n”.</td></tr><tr><td><i>n@x</i></td><td>List all users starting with the character “n” and ending with the character “x”.</td></tr><tr><td><i>n## ... #</i></td><td>List all users starting with the character “n” followed by up to seven digits.</td></tr><tr><td><i>?n@</i></td><td>List all users whose second character is “n”.</td></tr><tr><td><i>n?</i></td><td>List all two-character users starting with the character “n”.</td></tr><tr><td><i>?n</i></td><td>List all two-character users ending with the character “n”.</td></tr></table>	<i>@</i>	Specifies zero or more alphanumeric characters and denotes all members of the set.	<i>#</i>	Specifies one numeric character.	<i>?</i>	Specifies one alphanumeric character.	<i>n@</i>	List all users starting with the character “n”.	<i>@n</i>	List all users ending with the character “n”.	<i>n@x</i>	List all users starting with the character “n” and ending with the character “x”.	<i>n## ... #</i>	List all users starting with the character “n” followed by up to seven digits.	<i>?n@</i>	List all users whose second character is “n”.	<i>n?</i>	List all two-character users starting with the character “n”.	<i>?n</i>	List all two-character users ending with the character “n”.
<i>@</i>	Specifies zero or more alphanumeric characters and denotes all members of the set.																				
<i>#</i>	Specifies one numeric character.																				
<i>?</i>	Specifies one alphanumeric character.																				
<i>n@</i>	List all users starting with the character “n”.																				
<i>@n</i>	List all users ending with the character “n”.																				
<i>n@x</i>	List all users starting with the character “n” and ending with the character “x”.																				
<i>n## ... #</i>	List all users starting with the character “n” followed by up to seven digits.																				
<i>?n@</i>	List all users whose second character is “n”.																				
<i>n?</i>	List all two-character users starting with the character “n”.																				
<i>?n</i>	List all two-character users ending with the character “n”.																				
<i>listfile</i>	Destination of the attribute listing. The default is \$STDLIST, but the output may be redirected with the FILE command.																				

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** aborts execution. You must have AM (account manager) or SM (system manager) capability to execute this command.

Operation

The **LISTUSER** command generates an octal dump of the attributes currently assigned to users. Each entry is headed by "U = " and the user's name; it contains both user and system information. A translation of those bytes containing ASCII data appears to the right of each line, as shown in the sample below. Account managers may list only users in their own account, while system managers may specify all users in all accounts on the system.

```
U = COMGRAMS
041517 046507 051101 046523 060003 000613 000000 000000 COMGRAMS.....
020040 020040 020040 020040 050125 041040 020040 020040 .....PUB.....
000000 040226 000121 .....Q
```

The octal information is translated into a decimal representation below.

```
--00-- --01-- --02-- --03-- --04-- --05-- --06-- --07--
--08-- --09-- --10-- --11-- --12-- --13-- --14-- --15--
--16-- --17-- --18--
```



Each word or group of words is decoded in the following table:

Words	Content
0-3	User name
4-5	Capability
6-7	Local attributes
8-11	Password
12-15	Home group
16	Number of users logged-on
17	Maximum job priority
18	User entry in COMMAND.PUB.SYS

- The user name, password, and home group are eight-character names, padded from the right with blanks.
- Capability can be decoded by reference to the **-WHO** intrinsic. Refer to the *MPE V Intrinsic Reference Manual* (32033-90007).
- The maximum job priority is a numerical quantity; 150 (%226) for example, represents the CS subqueue.

LISTUSER

Example

To list the attributes of the user MGR in the account SOPRM, enter:

```
:LISTUSER MGR.SOPRM
```

```
U = MGR
```

```
046507 051040 020040 020040 077607 000713 000000 000000 MGR.....
```

```
020040 020040 020040 020040 050125 041040 020040 020040 .....PUB.....
```

```
000002 040226 000466 .....6
```

```
:
```

LISTVS

Produces a formatted listing of volume set definitions.

Syntax

```
LISTVS [vslist]  $\left[ \left\{ \begin{array}{l} ,0 \\ ,1 \\ ,2 \end{array} \right\} \right] [ ;listfile]$ 
```

Parameters

vslist Specifies the volume set definitions to be listed. This positional parameter has the form:

vsname[.*groupname*[.*acctname*]]

The characters “@”, “#”, and “?” can be used as wildcard characters in any position of the *vslist* parameter. The wildcard characters count toward the eight-character limit for user, group, account, and filenames. These wildcard characters have the following meanings:

- @ Specifies one or more alphanumeric characters. When used by itself, “@” denotes all members of the set.
- # Specifies one numeric character.
- ? Specifies one alphanumeric character.

These characters can be used as follows:

- n@ Lists all volume set definitions starting with the character “n”.
- @n Lists all volume set definitions ending with the character “n”.
- n@x Lists all volume set definitions starting with the character “n” and ending with the character “x”.
- n##### Lists all volume set definitions starting with the character “n” followed by seven digits.
- ?n@ Lists all volume set definitions whose second character is “n”.
- n? Lists all two-character volume set definitions starting with the character “n”.
- ?n Lists all two-character volume set definitions ending with the character “n”.

Depending on the placement of these characters in the *vslist* parameter, they can also be used to replace group and account names. If you do not specify names for the group or account, MPE V/E assumes the logon group and account. The default is “@” (lists all home volume sets in logon group).

LISTVS

- 0, 1, or 2 Displays information about volume sets. The 0 option displays volume set names and volume class names. Option 1 displays the information shown for 0 as well as volume names of volume set members and volume class members, and types of devices on which the volumes reside. Option 2 displays the information shown for 0 and 1. With option 2, if the device is mounted, then the *ldev* and the status "MOUNT" also appear. An asterisk (*) following a volume set indicates that the volume set is mounted. The default is 0.
- listfile* Actual file designator of the output file to receive listings. The default is \$STDLIST. If you do not use the default, this file must be named in a FILE command and backreferenced.

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** aborts execution. You must have use volumes (UV) or create volumes (CV) capability to execute this command.

Operation

The LISTVS command allows you to determine the configuration of a volume set and produce a formatted listing of volume set definition information.

The numeric parameter of the LISTVS command determines the amount of information displayed. If you omit it or if you specify 0, item 1 in the following list is displayed. If you specify 1, items 1 through 3 are displayed. If you specify 2, items 1 through 4 are displayed:

1. Volume set names and volume class names.
2. Volume set member volume names and types.
3. Volume class member volume names and types.
4. Logical device numbers (*ldev*) and status of the devices on which the volume set is mounted. This information is omitted for any volume set not currently mounted.

Examples

To display the all members, numbers, class names, and types of the volume set, along with the logical device number and status of the volume set, enter:

```
:LISTVS,2
```

```
ACCOUNT= TECHPUBS
```

```
GROUP= MAC
```

VOLSET	MEMBERS	TYPE	LDEV	STATUS
-----	-----	-----	----	-----
MAC*	MAC	HP7920	4	MOUNTED

```
:
```

To display the members and type of the volume set for all users and accounts, enter:

```
:LISTVS 0.0,1
ACCOUNT= TECHPUBS          GROUP= MAC

VOLSET      MEMBERS      TYPE
-----
MAC*        MAC          HP7920
:
```

To display the volume set numbers and volume class names, enter:

```
:LISTVS 0.0,0
ACCOUNT= TECHPUBS          GROUP= MAC

VOLSET
-----

MAC*
:
```

To display information on definitions for volume sets in your logon group and account, with the listing sent to a line printer, enter:

```
:FILE LIST;DEV=LP
:LISTVS 0,1;*LIST
```

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005)

LMOUNT

Performs a logical mount of a private volume set.

Syntax

```
LMOUNT vsname.group.account [ ;GEN=[genindex] ]
```

Parameters

<i>vsname</i>	Volume set name which specifies the mounting of a previously defined volume class or volume set.
<i>group.account</i>	Specifies the group and account under which the volume set was created.
<i>genindex</i>	A value from -1 to 32767 specifying which generation of the volume set is to be mounted (-1 allows any generation to be mounted). If omitted, the system does not check the generation version of the specified volume set.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only at the console *unless* it is distributed to users with the **ALLOW** command.

Operation

When you execute the **LMOUNT** command, all disk drives containing members of the specified volume set become “assigned”. Each volume set is logically attached to the drive until a **LDISMOUNT** command is executed, at which time the disk drive becomes “available”.

System users issue mount requests implicitly through their programs, or explicitly with a **MOUNT** command. Mount requests initiated by users cause the specified volume set to be linked into the system’s accounting structure at the group level. This is called “binding”. When the console **LMOUNT** command is executed, no such binding occurs.

If you enabled the private volumes facility with **VMOUNT ON,AUTO**, MPE V/E automatically attempts to satisfy the mount request; the **LMOUNT** succeeds if the specified volume class/set is physically connected to the system.

If you enabled the private volumes facility with **VMOUNT ON** (omitting the **AUTO** parameter), you must reply to your own mount request, even though the volume class/set may already be mounted and in use. If the specified volume class/set is not physically present, the system looks for available drives of the types required to satisfy the request. If sufficient drives are available, the system reserves them and prints a message telling you to mount the volume class/set on one or more logical devices, and to reply “**YES**” or “**NO**”. If some drives are not available, the system reserves those that are available, and prints a message listing the type and number of drives it still requires.

If you can free the required drive(s), reply “**YES**” after the drive is made available. If you reply “**NO**”, the original **LMOUNT** request is canceled.

Example

To mount a private volume set named DATABASE, in the PAYROLL group of the ACCTNG account, enter:

:LMOUNT DATABASE.PAYROLL.ACCTNG

Additional Discussion

Refer to the discussion of “Private Volumes” in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

LOG

Starts, restarts, or stops user logging.

Syntax

$$\text{LOG } \textit{logid} \quad \left\{ \begin{array}{l} \text{,RESTART} \\ \text{,START} \\ \text{,STOP} \end{array} \right\}$$

Parameters

<i>logid</i>	Logging identifier previously established with a user GETLOG command.
START	Initiates a logging process.
RESTART	Restarts a logging process.
STOP	Terminates a logging process.

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** has no effect on this command. You may execute it only at the console *unless* it is distributed to users with the ALLOW command.

Operation

The LOG command allows you to start, restart, or stop user logging. For further discussion of user logging, refer to the *MPE V System Operation and Resource Management Manual* (32033-90005) and to the *MPE V Intrinsics Reference Manual* (32033-90007).

You may change log files as you go, without the delay caused by executing a LOG command. You can use the CHANGELOG command to enable interactive log file changing, or you may use the AUTO parameter of the ALTLOG and GETLOG commands to enable automatic log file changing. Refer to the commands ALTLOG, CHANGELOG, GETLOG, and SHOWLOGSTATUS in this chapter.

Example

To start the logging process identified by *logid* LOGPROCX, enter:

:LOG LOGPROCX,START

=LOGOFF

Aborts all executing jobs and/or sessions and prevents any further logon.

Syntax

LOGOFF

Parameters

None.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may issue it only from the console.

Operation

The **=LOGOFF** command sets the job and session execution limits to 0 and aborts all jobs and sessions, including the session logged onto the system console. To enter subsequent console commands, you must log on to the console using **Ctrl A LOGON**.

When you execute this command, you leave the system in a "job/session inactive" state. However, since devices are still accepting jobs and/or sessions, job introduction is not disabled: system managers and system supervisors can still introduce and run HIPRI jobs. Spoolers still operate, but unless a user is able to initiate an HIPRI session at the system console, it is impossible to alter their operation.

Any pending requests that require a **=REPLY** from the system console must be satisfied before you issue a **=LOGOFF** command, or MPE V/E aborts the requests.

Example

To abort all executing jobs and/or sessions, enter:

: **Ctrl A**

=LOGOFF

16.53/25/ALL JOBS LOGGED-OFF



=LOGON

Enables the processing of jobs and/or sessions following an =LOGOFF command.

Syntax

=LOGON

Parameters

None.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may issue it only from the console.

Operation

The =LOGON command enables the processing of jobs and/or sessions following the execution of the =LOGOFF command. The =LOGON command re-establishes the job or session limits that were in effect before the =LOGOFF command was executed and allows non-HIPRI jobs and sessions to log on again.

Example

To enable the processing of jobs or sessions, enter:

: **Ctrl** **A**
=LOGON

MOUNT

Requests the system operator to mount a volume set.

Syntax

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

Parameters

- * or vsname** Specifies the volume set name. The asterisk (*) specifies the home volume set for the specified group and account, or the logon group and account if *groupname.acctname* is not specified. The *vsname* parameter is the volume set name, specifying a volume set name (previously defined with a NEWVSET command). The default is "*".
- groupname.acctname** Specifies the group and account under which the volume set was created. The default is logon group and account.
- genindex** A value from -1 to 32767 specifying which generation of the home volume set is to be mounted. A value of -1 indicates that any generation is permitted. If you omit this parameter, the system ignores the generation when attempting to satisfy the MOUNT request.

Use

You may issue this command from a session, a job or in BREAK.

You may not use it from a program. Pressing **BREAK** has no effect on this command.

Operation

The MOUNT command connects you to a specific volume set. In addition, this command allows you to specify that the volume set be treated as a home volume set; therefore, it is to be bound to the system's accounting structure with respect to a particular group. The requesting job is suspended until the mount is completed or rejected.

In addition, various user commands that give you access to your logon group's home volume set implicitly initiate mount requests if the volume set is not mounted already. An example using the BUILD command is:

```
:BUILD VFILE;DISC=500,10,1;REC=-80;DEV=VCLASS1
```

To issue a mount request programmatically you may issue an FOPEN call referencing a file residing on an unmounted volume set; this initiates an implicit mount request. An FOPEN mount remains in effect until you issue a corresponding FCLOSE intrinsic call. Use the programmatic request when a single step in a job or session requires a certain volume set. Refer to the *MPE V Intrinsics Reference Manual* (32033-90007) for a description of programmatic mounting.

MOUNT

Example

To request the system operator to mount volume set **MYSET**, generation index 43, enter:

:MOUNT MYSET;GEN=43

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005)

NEWACCT

Creates a new account and an account manager and PUB group associated with it.

Syntax

```
NEWACCT acctname, mgrname [ ;PASS=[password] ] [ ;FILES=[filespace] ]
[ ;CPU=[cpu] ] [ ;CONNECT=[connect] ] [ ;CAP=[capabilitylist] ]
[ ;ACCESS=[fileaccess] ] [ ;MAXPRI=[subqueueuname] ] [ ;LOCATTR=[localattribute] ]
[ ;VS=[volset:SPAN] ] [ ;USERPASS={ REQ } ] (1)
```

(1) The USERPASS parameter is available only if the HP Security Monitor (HP30392) has been installed.

Parameters

acctname Name to be assigned to the new account. This name must contain from one to eight alphanumeric characters, beginning with an alphabetic character.

mgrname Name of the account manager. This is always the first user created under the account. The account manager receives the following attributes:

User Password	None
Capability List	Same as the account capability list
Scheduling Priority	Same as the account maximum priority
Local Attribute	Same as the account local attributes
Home Group	PUB

The attributes of an account manager may be changed with the ALTUSER command after *mgrname* is defined. However, this user is never granted attributes greater than those assigned to the account.

password Account password, used for verifying logon access only. This password must contain from one to eight alphanumeric characters beginning with an alphabetic character. The default is that no password is assigned.

filespace The disk storage limit, in sectors, for the permanent files of the account. The maximum value you may define is 2,147,483,647 sectors. The default is unlimited file space.

cpu Limit on total CPU time, in seconds, for this account. This limit is checked only when a job or session is initiated. Thus, the limit never causes the job or session to abort. The maximum value you may define with NEWACCT is 2,147,483,647 seconds. The default is that no limit is assigned.

NEWACCT

connect Limit on total session connect time, in minutes, for the account. This limit is checked at logon, and when the job or session initiates a new process. The maximum value you may define is 2,147,483,647 minutes. The default is that no limit is assigned.

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

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

The default is **AM,AL,GL,SF,ND,IA,BA**.

fileaccess File access restrictions assigned to the account. The default is **R,L,A,W,X:AC**.

$$\left[\begin{array}{c} \left(\begin{array}{c} R \\ L \\ A \\ W \\ X \end{array} \right) \end{array} \right] [, \dots] : \left\{ \begin{array}{c} ANY \\ AC \end{array} \right\} [; \dots]$$

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

R = READ
L = LOCK (allows exclusive access to file)
A = APPEND (implicitly specifies L as well)
W = WRITE (implicitly specifies A and L as well)
X = EXECUTE

The user types are specified as follows:

ANY = Any user
AC = Member of this account only

The default is no security restrictions at the account level. Two or more modes may be specified if they are separated by commas. Both user types may be specified if they are separated by commas.

*subqueue**name* The name of the subqueue indicating the highest priority that can be requested by any process during any job or session in the account. This parameter is specified as AS, BS, CS, DS, or ES.

Caution



Processes capable of executing in the AS or BS subqueues can deadlock the system. Assigning non-priority system and user processes to these subqueues can prevent critical processes from executing. Exercise extreme caution when assigning processes to these subqueues.

localattribute The local attribute of the account, as defined at the installation site. This is a double-word bit map used to further classify accounts. When it is not part of standard MPE V/E security provisions, programmers may define local attributes (which are checked by the WHO intrinsic) to enhance their software's security. The default is double-word 0.

volset The volume set or class reference which, when fully qualified, is in the form *vcid.groupname.acctname*. *vcid* refers to a previously defined volume set or class.

SPAN Specifies that the *accountname* is to be inserted in the accounting directory of the specified volume set (*volset*). The specified volume set must already be physically mounted for the SPAN operation to succeed.

The associated account manager and PUB group are not created in the accounting directory of the specified volume set.

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

REQ USERPASS=REQ specifies that all users in the account are to have non-blank passwords. If you require user passwords, MPE V/E assigns the account manager a blank, expired password. The account manager must select a new password the first time the Manager logs on. It is available only if the HP Security Monitor has been installed.

OPT USERPASS=OPT specifies that users of the account may or may not have passwords. This is the default. It is available only if the HP Security Monitor has been installed.

NEWACCT

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have system manager (SM) capability to execute this command.

Operation

The **NEWACCT** command may only be executed by the system manager. The system manager is responsible for establishing the accounting structure best suited to the computer installation.

When a keyword is specified but its corresponding parameter is omitted (as in **;ACCESS=**), the default value for that keyword is assigned (in this case, **R, L, A, W, X:AC**). The default is also assigned when an entire keyword parameter group (such as **;ACCESS=fileaccess**) is omitted.

After the system manager has created accounts and their **PUB** groups, and has designated the account managers for those accounts, the new account managers may logon and redefine their own attributes and those of their **PUB** groups. Account managers may also define new users and groups. However, the capabilities and attributes the account managers assign to groups and users cannot exceed those assigned to the account itself by the system manager. For example, if the system manager does not assign the account **DS** capability, no users in the account are permitted **DS** capability (so they cannot use extra data segments).

The **PUB** group is initially assigned no password and the same capability class attributes, permanent file space limit, CPU limit, and connect time limit as the account. Its initial security allows **READ** and **EXECUTE** access to all users who successfully logon to the account, and **APPEND**, **WRITE**, **LOCK**, and **SAVE** access to account librarian (**AL**) and group users (**GU**) only. These access provisions are (**R,X:ANY;A,W,L,S:AL,GU**).

Example

To create an account with the account name **ACI**, and the account manager name **MNGR**, with all other parameters assigned by default, enter:

```
:NEWACCT ACI,MNGR
```

Additional Discussion

HP Security Monitor Users Guide (30392-90001)

NEWGROUP

Creates a new group within an account.

Syntax

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

Parameters

- groupname* The name of the new group, which must consist of one to eight alphanumeric characters, beginning with an alphabetic character.
- accountname* As system manager, you include the account name when you are logged onto an account other than the requested account.
- password* Group password, used for verifying logon access only. The default is that no password is assigned.
- capabilitylist* A list of capability-class attributes, consisting of any or all of the following: IA, BA, PM, MR, DS, or PH, where:

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

This list imposes a limit on program files belonging to the group. A capability cannot be assigned to the group if it has not been defined for the account in which the group resides. The default is IA, BA.

- filespace* Disk storage limit, in sectors, for the permanent files of the group. You cannot specify a *filespace* for a group that is greater than the limits currently defined for the group's account. If an account's *filespace* is changed later, group limits are automatically changed. The default is a storage limit equivalent to the account's *filespace*.

- cpu* Limit on total CPU time, in seconds, for the group. This limit is checked when a job or session is initiated, and does not cause the job or session to abort if it is exceeded during an active job or session. If the limit is exceeded, new jobs or sessions are prevented from initiating until the counter is reset with the RESETACCT command. The maximum value you may define with this command is 2,147,483,647 seconds. However, you cannot specify a CPU limit for a group if it exceeds the limit defined for the account. If an account's CPU limit is changed later, the group's limit is changed automatically. The default is the account CPU limit.

NEWGROUP

connect Limit on total session connect time, in minutes, for the group. This limit is checked at logon, and when the job or session initiates a new process. The maximum value you may define is 2,147,483,647 minutes, and it cannot exceed the limit defined for the account. If the account connect time is later changed, the group connect time is automatically changed. The default is the account connect limit.

fileaccess The restriction on file access pertinent to this group. The default is R,X:ANY;A,W,L,S:AL,GU for the public group (PUB), and R,A,W,L,X,S:GU for all other groups.

$$\left\{ \begin{array}{c} R \\ L \\ A \\ W \\ X \end{array} \right\} [, \dots] : \left\{ \begin{array}{c} ANY \\ AC \\ GU \\ AL \\ GL \end{array} \right\} [, \dots]$$

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

R = READ
L = LOCK (exclusive file access)
A = APPEND (implies L)
W = WRITE (implies A and L)
X = EXECUTE
S = SAVE

The user types are specified as follows:

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

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

volset The volume set or class reference which, when fully qualified, is in the form *vcid.groupname.accountname*. *vcid* refers to a previously defined volume set or class.

SPAN The SPAN option specifies that *groupname* is to be inserted in the accounting directory of the specified volume set (*volset*). The specified volume set must already be physically mounted for the SPAN operation to succeed.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have account manager (AM) capability to execute this command.

Operation

Account managers use the **NEWGROUP** command to create groups within their accounts and assign attributes to each. The attributes assigned to the group may not exceed those permitted the accounts themselves (defined when the system manager created the accounts). However, within account limits, the account manager may redefine the group and user attributes and capabilities, as well as those of the **PUB** group.

The **PUB** group is initially assigned no password and the same capability-class attributes, permanent file space limit, and CPU time limit as the account. Its initial security grants **READ** (R) and **EXECUTE** (X) access to all users (**ANY**) who successfully logon to the account. **APPEND** (A), **WRITE** (W), **LOCK** (L), and **SAVE** (S) access is assigned to the account librarian (AL) and group users (GU) only.

When you omit a keyword parameter (such as **;PASS=**) or keyword parameter group (such as **;PASS=password**) from the **NEWGROUP** command, the default value corresponding to that parameter is assigned.

Example

To create a new group named **GROUP1**, which is assigned all the default capabilities, enter:

```
:NEWGROUP GROUP1
```

Additional Discussion

MPE V Account Structure and Security (32033-90136)

NEWUSER

Defines a new user.

Syntax

```
NEWUSER username [ .acctname ] [ ;PASS=password ] [ ;CAP=capabilitylist ]  
[ ;MAXPRI=subqueuename ] [ ;LOCATTR=localattribute ]  
[ ;HOME=homegroupname ] [ ;USERPASS= $\left\{ \begin{array}{l} \text{REQ} \\ \text{OPT} \end{array} \right\}$  [ ,EXPIRED ] ] (1)
```

(1) The USERPASS parameter is available only if the HP Security Monitor has been installed.

Parameters

<i>username</i>	The name of the user. The name must consist of one to eight alphanumeric characters, beginning with an alphabetic character.
<i>accountname</i>	As system manager, you can include the account name when you are logged onto an account other than the requested account.
<i>password</i>	User password, used for verifying logon access only. The password must consist of one to eight alphanumeric characters, beginning with an alphabetic character. The default is that no password is assigned.
<i>capabilitylist</i>	List of capabilities, separated by commas, allowed this account. Each capability is denoted by a two-letter mnemonic, as follows:

System Manager	= SM
Account Manager	= AM
Account Librarian	= AL
Group Librarian	= GL
Diagnostician	= DI
System Supervisor	= OP
Network Administrator	= NA
Node Manager	= NM
Permanent Files	= SF
Access to non-sharable I/O devices	= ND
Use Volumes	= UV
Create Volumes	= CV
Use Communications Subsystems	= CS
User Logging	= LG
Process Handling	= PH
Extra Data Segments	= DS
Multiple RINs	= MR
Privileged Mode	= PM
Interactive Access	= IA
Local Batch Access	= BA
Programmatic Sessions	= PS

Capabilities assigned to the user via the ;CAP= parameter cannot exceed those assigned to the account. If the account's capabilities are altered, any capabilities removed from the account are also removed from the user. The user's capabilities are always a subset of the account's at logon. This prevents a user from being granted a capability not assigned to the account. Note that CV capability, which allows users to create private volumes, also confers UV capability, which lets them use those volumes. The default is IA, BA, ND, SF.

subqueue*name* The name of the highest priority subqueue that any job or session in the account can request for executing processes. The *subqueue**name* may be either AS, BS, CS, DS, or ES. The default is CS. The priority specified for the user in NEWUSER cannot be greater than that specified for the account.

The *subqueue**name* defined for the user is checked against the *subqueue**name* defined for the user's account at logon. The lower priority of the two is used as the maximum priority and restricts all processes of the job/session. Also, the priority requested by the user at logon is checked against the *subqueue**name* defined for that user, and the lower of these two values is granted.

Caution


Processes capable of executing in the AS or BS subqueues can deadlock the system. Assigning non-priority system and user processes to these subqueues can prevent the execution of critical system processes. Exercise extreme caution in assigning processes to these subqueues.

localattribute The local attribute of the user, as defined at the installation site. This is a double-word bitmap of arbitrary meaning, which may be used to classify users further. Although it is not part of standard MPE V/E security provisions, it is available to processes through the WHO intrinsic for use in the programmer's own security provisions. The NEWUSER command checks the local attributes of the user with those of the account. The default is double-word 0 (null).

homegroup-
name The name of an existing group to be assigned as the user's home group. If none is assigned, the user must specify a group when logging on. The default is that no home group is assigned.

REQ USERPASS=REQ requires that the user have a non-blank password. It is available only if the HP Security Monitor has been installed.

OPT USERPASS=OPT makes a user password optional. This is the default. It is available only if the HP Security Monitor has been installed.

EXPIRED The password expires immediately. The user cannot logon without selecting a new password. It is available only if the HP Security Monitor has been installed.



NEWUSER

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have account manager (AM) capability to execute this command.

Operation

The account manager uses the **NEWUSER** command to define an account member. When the user is defined, the account manager may also assign the user a password and capabilities, and limit the user's access to system resources. Parameters defining these values may also be omitted from the command line; in this case, the defaults are assigned to the user.

Example

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

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

Additional Discussion

HP Security Monitor Users Guide (30392-90001)

NEWVSET

Defines private volume sets and classes.

Syntax

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

Parameters

- vsname* Volume set name, consisting of one to eight alphanumeric characters, beginning with an alphabetic character. This name is also given to the master volume of the set. MPE V/E references *vsname* as *vsname.groupname.acctname*, where *groupname* and *acctname* are the logon group and account.
- vname:type* Names of members of the volume set and the types of devices on which the members reside. A maximum of eight volume names can be specified. The list of volume names must include the master volume set name. The type of disk to accommodate the volume must be specified with each *vsname*. The *type* can be any of these: HP 7902A, HP 7905, HP 7906, HP 7911, HP 7912, HP 7914, HP 7920, HP 7925, HP 7933, HP 7935, HP 7945, HP 9895, HP 7936, HP 7937, HP 7957, HP 7957B, HP 7958, HP 7958B, HP 7959B, HP 7961B, HP 7962B, or HP 7963B.
- vcname:vname* Volume class name, consisting of one to eight alphanumeric characters, beginning with an alphabetic character. Each volume class consists of a volume class name and an associated list of volume names (*vname*). The list of volume names must be a subset of the volumes composing the volume set, and one of the volume names must be that of the set's master volume.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have create volumes (CV) capability to execute this command.

Operation

The **NEWVSET** command defines private volume sets and classes, making known to MPE V/E the members of the volume set and their corresponding storage types. Once a volume set or volume class is defined, its definition is kept on disk in a volume set definition entry. A physical volume set is not created at the time the volume set is defined.

NEWVSET

Example

To define a volume set with *vsname* USRSET; with members USRSET (master volume and MM2, of *type* HP7920) and class (*vcname:vname*) CLASS1:USRSET,MM2, enter (all on one line):

:NEWVSETUSRSET;MEMBERS=USRSET:HP7920,MM2:HP7920;CLASS=CLASS1:USRSET,MM2

OPENQ

Opens the spool queue for a specified logical device or device class.

Syntax

$$\text{OPENQ} \left\{ \begin{array}{l} \textit{ldev} \\ \textit{devclass} \end{array} \right\}$$

Parameters

ldev The logical device number of the spooled device.

devclass The device class name of the spooled device; *devclass* must begin with a letter and may consist of up to eight alphanumeric characters.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You may execute it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The **OPENQ** command enables the operator to control the spool queue of a specified device or device class without affecting the operation of the spooler process. It also gives the operator access to spool queues for which no spooler or physical device exists.

Since spool files can be created faster than they are processed, you may want to issue a **SHUTQ** command to clear the backlog of files in the queue, and then re-open it with an **OPENQ** command when the queue is clear.

OPENQ also serves as an alternative to the **STOPSPPOOL** command. The **STOPSPPOOL** and the **SHUTQ** commands are documented in this chapter.

Example

To open the spool queue for logical device #6, enter:

:OPENQ 6

OUTFENCE

Defines the minimum priority that an output spool file needs in order to be printed.

Syntax

```
OUTFENCE outputpriority [ ;LDEV=ldev ]
```

Parameters

outputpriority A number between 1 and 14, inclusive. The larger numbers are more limiting than the smaller ones.

ldev The logical device number of an output device.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **(BREAK)** has no effect on this command. You can execute it only from the console *unless* it is distributed to users with the **ALLOW** command.

Operation

The **OUTFENCE** command controls the processing of all output spool files by establishing a numerical limit, or “fence”. This limit, along with each spool file’s *outputpriority*, determines whether a file is printed or deferred. Individual spool files that are **READY** are printed only if their *outputpriority* is higher than the current outfence. To prevent any spool files from being printed, set the outfence to 14. To defer a subset of spool files, set the outfence higher than the *outputpriority* of any spool file in the group.

You may wish to alter the printing priority of a single file without affecting the entire system. If that is the case, change the *outputpriority* of the specific spool file with the **ALTSPoolFILE** command. (For more information, refer to the discussion of **ALTSPoolFILE** in this chapter.)

Examples

To defer all output spool files except those waiting to be printed by logical device #6, usually configured as the system line printer, set the global outfence to 14 and the outfence of logical device #6 to 7, as shown below:

```
:OUTFENCE 14  
:OUTFENCE 7;LDEV=6
```

OUTFENCE

To display the new global *outputpriority* and the *outputpriority* of logical device #6, execute the **SHOWOUT** command, as illustrated in the example below. The summary statistics at the bottom of the listing show the new outfence immediately, but individual entries for each output spool file may not. Once the currently active spool file is finished, however, no files directed towards a device other than *ldev=6* become active.

:SHOWOUT

DEV/CL	DFID	JOBNUM	FNAME	STATE	FRM	SPACE	RANK	PRI	#C
6	#0999	#J19	\$STDLIST	OPENED		512		8	1
6	#01030	#S77	EDLIST	OPENED		512		8	1
SLOWLP	#01029	#S71	OUT	READY		232	1	7	1
20	#01001	#S60	\$STDLIST	OPENED					
11	#01022	#S33	GALLIST	READY		768	1	7	1

11 FILES:

- 1 ACTIVE
- 1 READY; INCLUDING 1 SPOOFLES, 0 DEFERRED
- 9 OPENED; INCLUDING 2 SPOOFLES
- 0 LOCKED; INCLUDING 0 SPOOFLES
- 4 SPOOFLES: 2024 SECTORS

OUTFENCE = 14

OUTFENCE = 7 FOR LDEV 6

To reset the outfence for all output spool files, enter:

:OUTFENCE 6

PARTBACKUP

Performs a backup of MPE V/E and all of the files that have been modified since the last full backup.

Syntax

```
PARTBACKUP [ dumptape ] [ , auxlistfile ]
```

Parameters

<i>dumptape</i>	Specifies the magnetic tape destination for the backup. If <i>dumptape</i> is specified, it must be backreferenced to a previous FILE command or file equated to the formal file designator DUMPTAPE . The default is TAPE . If there is no device class TAPE , the device class CTAPE is used.
<i>auxlistfile</i>	Actual file designator of the output file or device to which all listings requested during the execution of the backup process are written. The default is \$STDLIST .

Use

You may issue this command from a session or job , but not in **BREAK**. Pressing **BREAK** suspends execution. You require system supervisor (OP) capability to use it.

Operation

This command starts the backup process and copies MPE V/E and all the files that have been modified since the last full backup. The commands **PARTBACKUP** and **FULLBACKUP** provide an alternative to going through the **SYSDUMP** dialog. The only user interaction required with these commands is mounting tapes or serial discs and replying to any mount requests. These replies are unnecessary, however, if the devices are configured as **AUTOREPLY**.

During the backup, **PARTBACKUP** displays messages at one- minute intervals to report its progress; for example:

```
STORE OPERATION IS 27% COMPLETE
```

Examples

In the following example, the first **FILE** command defines **DUMP** as a magnetic tape file. The second **FILE** command defines **LIST** as a line printer. The **TAPE** and **LP** parameters are device class names arbitrarily defined during the previous system configuration.

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

The following is displayed at the console:

```
?11:08/#S278/89/ LDEV# FOR "DUMP" ON TAPE (NUM)?
```

In the following example, a file equation with the formal file designator of DUMPTAPE had already been entered (that is, FILE DUMPTAPE;DEV=SYSTAPE). As a result, you may omit the first parameter, and the backup goes to the device class SYSTAPE:

:PARTBACKUP,*LIST

The following is displayed at the console:

```
?11:08/#S278/89/ LDEV# FOR "DUMPTAPE" ON SYSTAPE (NUM)?
```

In the following example, the *dumptape* is not specified, and there is no file equation for DUMPTAPE. A temporary file equation (FILE DUMPTAPE;DEV=TAPE) is used.

:PARTBACKUP

The following is displayed at the console:

```
?11:08/#S278/89/ LDEV# FOR "DUMPTAPE" ON TAPE (LDEV)?
```

PASCAL

Compiles a PASCAL program. PASCAL/3000 is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
PASCAL [ textfile ] [ , [ uslfile ] [ , listfile ] ] [ ; INFO=quotedstring ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is PASTEXT. The default is \$STDIN.
<i>uslfile</i>	Actual file designator of the USL file to which the object code is stored. This can be any binary output file with a file code of USL or 1024. Its formal file designator is PASUSL. If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS. If entered, this parameter indicates that the USL file was created in one of four ways: ■ By using the MPE V/E SAVE command to save the default USL file \$OLDPASS created during a previous compilation. ■ By building the USL with the MPE V/E Segmenter -BUILDUSL command. Refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011). ■ By creating a new USL file and specifying the MPE V/E BUILD command with a file code of USL or 1024. ■ By specifying a nonexistent <i>uslfile</i> parameter, thereby creating a permanent file of the correct size and type.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is PASLIST. The default is \$STDLIST.

Note



The formal file designators used in this command (PASTEXT, PASUSL, and PASLIST) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the FILE command.

<i>quotedstring</i>	A sequence of characters between two single quotation marks or between two double quotation marks. You may use the delimiter as part of the string provided that the delimiter appears twice. Any occurrence of two single quotation marks in a row, or two double quotes in a row, is considered part of the string and not the terminating delimiter. <i>INFO=quotedstring</i> is used in the PASCAL programming language to pass initial compiler options to a program. PASCAL/3000 brackets the <i>quotedstring</i> with dollar signs and places it before the first line of source code in the text file. For more information, refer to the RUN command in this chapter.
---------------------	--

Use

You may issue this command from a session or a job. You may not use it from a program or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **PASCAL** command compiles a PASCAL program and stores the object code in a user subprogram library (USL) file on disk. If *textfile* is not specified, MPE V/E expects the source program to be entered from your standard input device. If you do not specify a *listfile*, MPE V/E sends the program listing to your standard list device and identifies it by the formal file designator, **PASLIST**.

If you create the USL prior to compilation, you must specify a file code of USL or 1024. If you omit the *uslfile* parameter, the object code is saved in the temporary file domain as **\$OLDPASS**. To keep it as a permanent file, you must save **\$OLDPASS** under another name.

Examples

The following example compiles a PASCAL program entered from the standard input device and stores the object code in the USL file **\$OLDPASS**. The listing is then sent to the standard list device.

```
:PASCAL
```

The next example compiles a PASCAL program contained in the disk file **PASCSRC**, and stores the object code in the USL file **PASCOBJ**. The program listing is stored in the disk file **LISTFILE**.

```
:PASCAL PASCSRC,PASCOBJ,LISTFILE
```

Additional Discussion

PASCAL/3000 Reference Manual (32106-90001)
MPE Segmenter Reference Manual (30000-90011)

PASCALGO

Compiles, prepares, and executes a PASCAL/3000 program. PASCAL is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
PASCALGO [textfile][,listfile] [;INFO=quotedstring]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is PASTEXT. The default is \$STDLIST.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is PASLIST. The default is \$STDLIST.

Note



The formal file designators used in this command (PASTEXT and PASLIST) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the FILE command.

<i>quotedstring</i>	A sequence of characters between two single quotation marks or between two double quotation marks. You may use the delimiter as part of the string provided that the delimiter appears twice. Any occurrence of two single quotation marks in a row, or two double quotes in a row, is considered part of the string and not the terminating delimiter. <i>INFO=quotedstring</i> is used in the PASCAL programming language to pass initial compiler options to a program. PASCAL/3000 brackets the <i>quotedstring</i> with dollar signs and places it before the first line of source code in the text file. For more information, refer to the RUN command in this chapter.
---------------------	--

Use

You may issue this command from a session or job. You may not use it from a program, or in BREAK. Pressing **BREAK** suspends execution.

Operation

The PASCALGO command compiles, prepares, and executes a PASCAL program. If *textfile* is omitted, MPE V/E expects input from your standard input device. If you do not specify *listfile*, MPE V/E sends the program listing to the formal file designator PASLIST (the default is \$STDLIST).

The USL file created during the compilation is the temporary file defined by the system, called \$OLDPASS, which is passed directly to the Segmenter. It can only be accessed if you do not use the default for *progfile*. This is because the Segmenter also uses \$OLDPASS to store the prepared program segments, overwriting any existing temporary file of the same name.

Examples

To compile, prepare, and execute a PASCAL program entered from your standard input device, with the program listing sent to your standard list device, enter:

:PASCALGO

To compile, prepare, and execute a PASCAL program from the disk file PASC SRC and send the program listing to the file LISTFILE, enter:

:PASCALGO PASC SRC,LISTFILE

Additional Discussion

PASCAL/3000 Reference Manual (32106-90001)
MPE Segmenter Reference Manual (30000-90011)



PASCALPREP

Compiles and prepares a PASCAL/3000 program. PASCAL is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
PASCALPREP [textfile] [, progfile] [, listfile] [ ; INFO=quotedstring ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is PASTEXT. The default is \$STDIN.
<i>progfile</i>	Actual file designator of the program file to which the prepared program segments are written. You get the best results by omitting the <i>progfile</i> parameter. When <i>progfile</i> is omitted, the MPE V/E Segmenter creates the program file, which is stored in the temporary file domain as \$OLDPASS. If you do create your own program file, you must do so in one of two ways: ■ By using the MPE V/E BUILD command, and specifying a file code of 1029 or PROG and a <i>numextents</i> value of 1. The PREP command then uses this file. ■ By specifying a nonexistent file in the <i>progfile</i> parameter, in which case a temporary job/session file of the correct size and type is created.
<i>listfile</i>	Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is PASLIST. The default is \$STDLIST.

Note



The formal file designators used in this command (PASTEXT and PASLIST) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the FILE command.

<i>quotedstring</i>	A sequence of characters between two single quotation marks or between two double quotation marks. You may use the delimiter as part of the string provided that the delimiter appears twice. Any occurrence of two single quotation marks in a row, or two double quotes in a row, is considered part of the string and not the terminating delimiter. <i>INFO=quotedstring</i> is used in the PASCAL programming language to pass initial compiler options to a program. PASCAL/3000 brackets the <i>quotedstring</i> with dollar signs and places it before the first line of source code in the text file. For more information, refer to the RUN command in this chapter.
---------------------	--

Use

You may issue this command from a session or job. You may not use it from a program or in BREAK. Pressing **BREAK** suspends execution.

Operation

The PASCALPREP command compiles and prepares a PASCAL program into a program file on disc. If you do not specify *textfile*, MPE V/E expects input from the current input device. If you do not specify *listfile*, MPE V/E sends the listing output to the formal file designator PASLIST (the default is \$STDLIST). The USL file \$OLDPASS, created during compilation, is a temporary file passed directly to the Segmenter. You may access it only if you do not use the default for *progfile*. This is because the Segmenter also uses \$OLDPASS to store the prepared program segments, overwriting any existing temporary file of the same name.

Examples

The following example compiles and prepares a PASCAL program entered through your standard input device and stores the prepared program segments in the file \$OLDPASS. The listing is printed on your standard list device.

```
:PASCALPREP
```

To compile and prepare a PASCAL source program from the source file PASC SRC, storing it in PASC PROG, and sending the listing to your standard list device, enter:

```
:PASCALPREP PASC SRC,PASC PROG
```

Additional Information

PASCAL/3000 Reference Manual (32106-90001)

MPE Segmenter Reference Manual (30000-90011)

PASSWORD

Allows you to establish or change your own password.

Syntax

PASSWORD

Parameters

None.

Use

You can use this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **PASSWORD** command has two purposes: to allow you to create or change your own user password, and to ensure that user passwords are private. You no longer need account manager or system manager capabilities to change your own password.

A user password must consist of one to eight alphanumeric characters, beginning with an alphabetic character. To enhance system security, the system manager may configure a system-wide minimum password length from 0 to 8. The minimum length requirement does not apply to existing passwords. If you select a new password that does not meet the length requirement, you see the following message:

MINIMUM PASSWORD LENGTH IS ! CHARACTERS. (CIERR 763)

If your password length is incorrect, see your system manager for assistance. The default minimum length is 0.

To execute the **PASSWORD** command, you must enter the old user password before a new user password prompt appears. If you enter the old user password incorrectly, you get an error and the command is aborted. When entering the new user password, you are requested to enter the password a second time for verification. If the second password differs from the first, you get an error and the command is aborted.

The passwords you type (old and new) are not displayed on the terminal screen. If the old password is entered incorrectly, you get an error message, as shown below:

PASSWORD WAS NOT CHANGED.

Example

To change a user password from “SMITTY” to “JONES”, you would enter:

:PASSWORD

ENTER OLD USER PASSWORD: SMITTY

ENTER NEW USER PASSWORD: JONES

ENTER NEW USER PASSWORD AGAIN: JONES

PASSWORD WAS CHANGED SUCCESSFULLY.

Additional Discussion

MPE V Account Structure and Security (32033-90136)

PREP

Prepares a program from a user subprogram library (USL) file into a program file.

Syntax

```
PREP uslfile,progfile [ ;ZERODB ] [ ;CAP=capabilitylist ] [ ;PMAP ] [ ;RL=filename ]  
[ ;MAXDATA=segsz ] [ ;PATCH=patchsz ] [ ;STACK=stacksz ] [ ;NOSYM ] [ ;DL=dlsz ]  
[ { ;FPMAP } ]  
[ { ;NOFPMAP } ]
```

Parameters

<i>uslfile</i>	Actual designator of the user subprogram library (USL) file into which the program is compiled.
<i>progfile</i>	Actual designator of the program file onto which prepared program segments are written. This can be any binary output file created in one of two ways: ■ By creating a new file with the MPE V/E BUILD command using a file code of PROG or 1029 and one extent. ■ By specifying a nonexistent file in the <i>progfile</i> parameter, in which case a file of the correct size and type is created. This file is a temporary job file.
ZERODB	Request to initialize to zero the initially defined, user-managed (DL-DB) area of the stack, and uninitialized portions of the DB-Q (initial) area. The default is that these areas are not affected.
PMAP	Request to produce a descriptive listing of the prepared program to a file whose formal file designator is SEGLIST. If no FILE command is found referencing SEGLIST, the listing is produced on \$STDLIST. The default is no listing.
<i>segsz</i>	Maximum permitted stack area (Z-DL) in words. You should include this parameter when you expect the size of DL-DB or Z-DB areas to be changed during program execution. The default is that MPE V/E assumes that these areas will not change.
<i>stacksz</i>	Size of initial local data area (Z-Q initial) stack, in words. This value, if specified, must be between 511 and 32,767 words. This parameter overrides the default <i>stacksz</i> estimated by the MPE V/E Segmenter.
<i>dlsz</i>	DL-DB area to be initially assigned to stack. This area is of interest mainly in programmatic applications. The DL-DB area is always rounded upward so that the distance from the beginning of the stack data segment to the DB-address is a multiple of 128 words. The default is estimated by the MPE V/E Segmenter.

<i>caplist</i>	Capability-class attributes associated with a program, specified as two-character mnemonics. If more than one mnemonic is specified, each must be separated from its neighbor by a comma. The mnemonics are: IA = Interactive Access BA = Local Batch Access PH = Process Handling DS = Data Segment Management MR = Multiple Resource Management PM = Privileged Mode Note that you can specify only those capabilities as assigned by the account manager or system manager. The default is IA and BA.
<i>filename</i>	Actual file designator of the relocatable library (RL) file to be searched to satisfy external references during preparation. This can be any permanent binary file of type RL. It need not belong to your logon group, nor have a reserved, local name. This file, to which you must have READ and LOCK access, yields a single segment that is incorporated into the segments of the program file. For more information, refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011). The default is that no library is searched.
<i>patchsize</i>	Specifies the size of the patch area. This size applies to all segments within the program file.
NOSYM	Suppresses the symbolic debug option. Refer to the <i>HPToolset Reference Manual</i> (32350-90001).
FPMAP or NOFPMAP	Includes or excludes the internal PMP information. FPMAP is a request to have internal PMP information included in the program. NOFPMAP excludes PMP information from the program when the system FPMAP or job/session FPMAP is on. If the symbolic debug option is invoked, the default is FPMAP. Otherwise the default is NOFPMAP.

Use

You may issue this command from a session or job. You may not use it from a program, or in **BREAK**. Pressing **(BREAK)** suspends execution.

Operation

The **PREP** command prepares a compiled source program for execution. Unless you prepare the program into a previously-created program file, **PREP** creates a temporary program file for you. It is a good idea to specify a nonexistent program file when you issue the **PREP** command. This way, MPE V/E creates a file of the optimum size and characteristics, as shown in the "Example".

The system prepares a compiled program by searching a relocatable library (RL) to satisfy references to external procedures required by the program. When the program is prepared, such procedures are linked to the program in the resulting program file. To use a relocatable library (RL) via the **PREP** command, you require **READ** and **LOCK** access to the file.

PREP

Caution



Use care when you prepare a USL file with symbolic debug. If you have temporary files named `SEGTMP00` and `SEGTMP01` in your group, save them as permanent files before entering the **PREP** command, otherwise you lose them. The Segmenter uses `SEGTMP00` and `SEGTMP01` when it prepares a USL file with symbolic debug, and these temporary files are purged or modified afterwards.

Example

To prepare a program from the USL file `USLX` and store it in the currently nonexistent program file `PROGX`, enter:

```
:PREP USLX,PROGX  
:SAVE PROGX
```

You obtain the best results by allowing the MPE V/E Segmenter to create your program file; however, you can create your own program file in the permanent file domain by using the **BUILD** command. When you do this, you must be sure to specify a file code of `PROG` or `1024` and a *numextents* parameter value of 1. The following **BUILD** command creates such a file, which is then used by the **PREP** command.

```
:BUILD PROGX;CODE=PROG;DISC=,1  
:PREP USLX,PROGX
```

To prepare a program from the USL file named `USLZ` and store it in a program file named `PROGZ`, list the prepared program, assign a *stacksize* of 511 words, and assign local batch access (BA) capability only for the program, enter:

```
:PREP USLZ,PROGZ;PMAP;STACK=511;CAP=IA,BA,PH,DS
```

Additional Discussion

For a complete tutorial, refer to the *MPE Segmenter Reference Manual* (30000-90011)

PREPRUN

Prepares and executes a compiled program.

Syntax

```
PREPRUN uslfile [ , entrypoint ] [ ;NOPRIV ] [ ;RL=filename ] [ ;PMAP ] [ ;NOCB ]
[ ;DEBUG ] [ ;INFO=quotestring ] [ ;LMAP ] [ ;STDIN= { *formaldesig
fileref
$NULL
} ] [ ;ZERODB ]
[ ;STDLIST= { *formaldesig
fileref [ ,NEW ]
$NULL
} ] [ ;MAXDATA=segsizesize ] [ ;PATCH=patchsize ]
[ ;PARM=parameternum ] [ ;NOSYM ] [ ;STACK=stacksize ] [ { ;FPMAP
;NOFPMAP } ]
[ ;DL=dlsizesize ] [ ;LIB= { G
P
S
} ] [ ;CAP=capabilitylist ]
```

Parameters

<i>uslfile</i>	Actual file designator of the USL file to which the program has been compiled.
<i>entrypoint</i>	Contains a character string terminated by a blank, specifying the entry point, or label, in the program where execution is to begin. The <i>entrypoint</i> parameter may be the primary entry point or any secondary entry point in the program's outer block. The default is the primary entry point.
NOPRIV	Declaration that the program segments are placed in non-privileged mode. This parameter is for programs prepared with privileged mode (PM) capability and makes them accessible to non-privileged users. Normally, program segments containing privileged instructions are executed in privileged mode only if the program was prepared with privileged mode capability class. (A program containing legally compiled privileged code and placed in non-privileged mode may abort when you attempt to execute it.) If you specify NOPRIV, all segments are placed in non-privileged mode. (Library segments are not affected because their mode is determined independently.) The default is that segments of a privileged mode program remain in privileged mode.
PMAP	Request to produce a descriptive listing of the prepared program to a file whose formal file designator is \$SEGLIST. If \$SEGLIST is not found in a FILE command, the listing is produced on the current list device. The default is no listing.

PREPRUN

DEBUG	Request to issue a DEBUG call before the first executable instruction of the program. Unless you have READ and EXECUTE access to the program file, this parameter is ignored. If privileged mode (PM) capability has been assigned, you are put into privileged mode DEBUG . If not, you are put into user mode DEBUG . The default is that the DEBUG call is not issued.
LMAP	Request to produce a descriptive listing of the allocated program to a file whose formal file designator is LOADLIST . If no FILE command referencing LOADLIST is found, the listing is produced on \$STDLIST . The default is no listing.
ZERODB	Request to initialize to zero the initially defined user managed (DL-DB) area, and uninitialized portions of the DB-Q (initial) area. The default is that these areas are not affected.
<i>segsiz</i>	Maximum permitted stack area (Z-DL) in words. This parameter should be included when you expect the size of DL-DB or Z-DB areas to change during program preparation or execution. Regardless of what you specify, MPE V/E may change the <i>segsiz</i> to accommodate table overflows. If you prepare your program with a <i>segsiz</i> less than the configured minimum, the value is rounded up to the minimum or the amount needed by the program (as calculated by the Segmenter). The maximum actual <i>segsiz</i> permitted a program is 32,767 words. You may prepare your program with a <i>segsiz</i> larger than necessary provided that this maximum is not exceeded. If the specified <i>segsiz</i> does exceed the maximum, it is rounded down to 32,767 words.
<i>parameternum</i>	An integer containing a parameter to be passed to the new program (accessed through Q-4 of the outer block).
<i>stacksiz</i>	Size of local data area, Z-Q (initial), in the stack, in words. If it is specified, this value must be between 511 and 32,767 words. The default is estimated by the MPE V/E Segmenter.
<i>dlsize</i>	DL-DB area to be initially assigned to stack. Due to system logging considerations, the DL-DB area is always rounded upward, so that the distance from the beginning of the stack data segment to the DB-address is a multiple of 128 words. The value you specify must be between -1 and 32,767 words. The default is estimated by the MPE V/E Segmenter.
G,P,or S	Searches the segmented procedure libraries of the program file's group and account. The G option searches the group library, account library, then the system library. The P option searches the account library then system library. The S option searches the system library for external references to segmented procedures. The default is S .

<i>capabilitylist</i>	Capability class attributes associated with the program, specified in two-character mnemonics. If more than one mnemonic is specified, each must be separated from its neighbor by a comma. The mnemonics are: IA = Interactive Access BA = Local Batch Access PH = Process Handling DS = Data Segment Management MR = Multiple Resource Management PM = Privileged Mode You can specify only those attributes that you were assigned by the account manager or the system manager. The default is IA and BA.
<i>filename</i>	Actual file designator of the relocatable library (RL) file to be searched to satisfy external references during preparation of the program. This can be any permanent file of type RL, to which you must have READ and LOCK access. It need not belong to the logon group, nor does it require a reserved, local name. This file yields a single segment that is incorporated into the segments of the program file. Refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011) for a description of RL files. The default is that no library is searched.
NOCB	Request that the file system not use stack segment (PCBX) for its control blocks, even if sufficient space is available. This lets you expand your stack (with the DLSIZE or ZSIZE intrinsics) to the maximum possible limit at a later time. It does, however, cause the file management system to operate more slowly for this program.
<i>quotedstring</i>	A sequence of characters between two single quotation marks or two double quotation marks. You may use the delimiting character as part of the string provided that the delimiter appears twice. Any occurrence of two single quotes, or two double quotes in a row, is considered part of the string and not the terminating delimiter. ;INFO=quotedstring is used in some programming languages (e.g. COBOLII, Pascal) to pass compiler options to a program. These options appear before the first line of source code in the text file. For more information, refer to the RUN command in this chapter.
STDIN	This parameter allows the user to specify the file to be used as \$STDIN by the program being executed. If you omit it, or if you do not specify anything after the “equals” sign (for example, ;STDIN=), then STDIN defaults to the job or session’s standard input device.

PREPRUN

You may use one of the following subparameters with **STDIN=**:

**formaldesig* The formal file designator for a file previously specified in a file equation.

fileref The name of an existing permanent disk file.

\$NULL The actual file designator of a system-defined file that is always treated as an empty file.

When a program references **\$NULL** as **\$STDIN** and tries to access the file, the program receives only an end-of-file indicator. When referenced by a program as **\$STDLIST**, the associated **WRITE** request is accepted by MPE V/E, but no physical output is actually generated. Thus, **\$NULL** can be used to discard unneeded output from an executing program.

STDLIST You can use this parameter to specify the file to be used as **\$STDLIST** by the program being executed. If you omit it, or do not specify anything after the “equals” sign (for example, **;STDLIST=**), then **STDLIST** defaults to the job or session’s standard list device. This parameter has the same subparameters as **STDIN**, but you may also specify the keyword **NEW**.

NEW The name to be assigned to a temporary job or session disk file created with the system defaults. The system default values for the new file are fixed-length ASCII 132-byte records with a maximum file size of 1023 records.

patchsize Specifies the size of the patch area. This definition applies to all segments in the program file. The value you specify must be between -1 and 16,380 words.

NOSYM Suppresses the symbolic debug option. Refer to the *HPToolset Reference Manual* (32350-90001) for more information.

FPMAP or **NOFPMAP** Includes or excludes the internal PMAP information. **FPMAP** is a request to have internal PMAP information included in the program. **NOFPMAP** excludes PMAP information from the program when the system **FPMAP** or job/session **FPMAP** is on. If the symbolic debug option is invoked, the default is **FPMAP**. Otherwise the default is **NOFPMAP**.

Use

You may issue this command from a session or job. You may not use it from a program, or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **PREPRUN** command prepares and executes a program compiled in a USL file. Both relocatable (RL) and segmented (SL) libraries are searched during the preparation process to satisfy external references.

The USL file created during compilation is a temporary file defined by the system, called **\$OLDPASS**, which is passed directly to the Segmenter. It can be accessed only if you do not use the default for *progfile*. This is because the Segmenter also uses the file **\$OLDPASS** to store the prepared program segments, overwriting any existing temporary file of the same name.

Caution

Use care when you prepare and execute a USL file with symbolic debug. If you have temporary files named SEGTMP00 and SEGTMP01 in your group, save them as permanent files before entering the PREPRUN command, otherwise you lose them. The Segmenter uses SEGTMP00 and SEGTMP01 when it prepares and executes a USL file with symbolic debug, and these temporary files are purged or modified afterwards.

**Examples**

To prepare and execute a program from the USL file XUSL, with no special parameters declared, enter:

```
:PREPRUN XUSL
```

To obtain a descriptive listing of the prepared program and a listing of the allocated (loaded) program, enter:

```
:PREPRUN XUSL;PMAP;LMAP
```

To prepare and execute a program from the USL file UBASE that begins execution at the entry point RESTART, that has a stack size of 800 words, and that searches an RL file named LIBA, enter:

```
:PREPRUN UBASE,RESTART;STACK=800;RL=LIBA
```

The following example prepares and runs a program with \$STDIN set to the existing disk file INPUT. \$STDLIST is set to the line printer:

```
:FILE LPFILE;DEV=LP  
:PREPRUN TESTPROG;MAXDATA=10000;STDIN=INPUT;STDLIST=*LPFILE
```

The next example also uses the STDIN= and STDLIST= parameters to prepare and run a program. This time, a file equation is used to set \$STDIN to INPT, and to set STDLIST to the temporary disk file RESULTS. This file is automatically created by the RUN command.

```
:FILE INFILE=INPT,OLD;  
:PREPRUN TESTPROG;DEBUG;STDIN=*INFILE;STDLIST=RESULTS,NEW
```

The following example of the PREPRUN command uses the INFO= parameter to pass a string to the program:

```
:PREPRUN MYPROG;MAXDATA=2000;INFO="A test with ""and"" characters"
```

The delimiting character is doubled within the string so that it appears on the printout as follows:

```
A test with "and" characters
```

Additional Discussion

For a full tutorial, refer to the *MPE Segmenter Manual* (30000-90011)

PTAPE

Reads a paper tape without X-OFF control.

Syntax

PTAPE *filename*

Parameter

filename Name of the existing ASCII file on disk, to which input data is to be written from a paper tape. Normally, this is a file with variable length records; the record size specified must be large enough to contain the largest paper tape record.

Use

You may issue this command from a session, program, or in **BREAK**. You may not use it in a job. Pressing **BREAK** has no effect on this command.

Operation

The **PTAPE** command lets programs read input from a terminal that is coupled to a paper tape reader/punch when the X-OFF control character is not punched on the tape. (The X-OFF character on a tape turns off the tape reading mechanism. This character normally appears following the carriage return/line feed character combination used to delimit each record.) The **PTAPE** command transfers all data from the tape to an ASCII file (named in the *filename* parameter) on disk. It stores all records to the disk file in the order read from the tape, but deletes all carriage return and line feed characters. (If the tape does contain X-OFF characters, the operation also deletes these. The X-OFF characters are not ignored by the tape reader, however, and so you must restart the reader after each X-OFF is read.) The input terminates when the **CTRL** Y character is encountered on the tape or entered from the terminal, and control returns to the keyboard. The input data can now be read from the disk file by any program.

The **PTAPE** command is required only for input from HP tape readers associated with terminal keyboards.

You cannot use the **PTAPE** command to copy tapes containing more than 32,767 bytes of information. If you attempt to copy such tapes, MPE V/E reads only the first 32,767 bytes on the tape and then terminates the operation.

Example

To copy input from a paper tape to a disk file named **TAPEFILE**, you must first create an appropriate file with the MPE V/E **BUILD** command, and then issue the **PTAPE** command:

```
:BUILD TAPEFILE;REC=80,3,V,ASCII  
:PTAPE TAPEFILE
```

PURGE

Deletes a file from the system.

Syntax

```
PURGE filereference [ ,TEMP ]
```

Parameters

filereference Actual file designator of the file to be deleted, in the following format:

filename [/ *lockword*] [. *groupname* [. *acctname*]]

To delete the file, you must have **WRITE** access to it.

TEMP Indicates that the file is a temporary file in the job/session temporary file domain. You must enter this parameter to delete a temporary file. The default is that a permanent file is assumed.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **PEURGE** command deletes a disk file from the system. If the volume set on which the file resides is not mounted, this command generates an implicit mount request. If a mount follows, the purge is done.

If the file does not exist in the specified domain (temporary or permanent), the following message is displayed:

```
FILE filename NOT FOUND, NO PURGE DONE. (CIWARN 383)
```

To purge a KSAM file, run **KSAMUTIL.PUB.SYS** using the **>PURGE** subcommand. Purging KSAM files with the MPE V/E **PURGE** command is not recommended.

Examples

To delete the permanent file **PFILE**, enter:

```
:PURGE PFILE
```

To delete the temporary file **TFILE**, enter:

```
:PURGE TFILE,TEMP
```

Additional Discussion

MPE File System Reference Manual (30000-90236)

PURGEACCT

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

Syntax

```
PURGEACCT acctname [;VS=volset]
```

Parameters

acctname Name of the account to be deleted. This name must contain from one to eight alphanumeric characters, beginning with an alphabetic character.

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

vcsid.groupname.acctname

The *vcsid* parameter refers to a previously defined volume set or volume class. When the *volset* is specified, the volume set or volume class must be mounted or the PURGEACCT command will fail. When *volset* parameter is specified, the account is removed from the volume set directory, and not the system directory.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have system manager (SM) capability to execute this command.

Operation

The system manager uses the PURGEACCT command to eliminate an entire account from the system. When PURGEACCT is executed during a session, MPE V/E displays a verification request to ensure that the wrong account is not deleted accidentally. Respond with "YES" or "NO" to the message:

```
ACCT acctname TO BE PURGED?
```

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

The PURGEACCT command removes every user not currently logged on and every group or file not in use. The order in which entries are purged is: users, volume set definitions, files, groups, and finally the account. If the command is executed while the account is in use, the account remains on the system: active users, groups, and files are not purged from the account. To completely purge an account, you must execute PURGEACCT when the account is inactive.

Caution

Do not attempt to purge the SYS account. It cannot be completely purged, but you can destroy critical files by attempting to do so. If you execute PURGEACCT SYS, all groups except PUB are purged, all users except the system manager are purged, and all inactive files and system files in the PUB group are purged.

Example

To remove an account named ACCT1, enter:

```
:PURGEACCT ACCT1
```

```
ACCOUNT ACCT1 TO BE PURGED? (YES/NO) YES
```

```
:
```

PURGEGROUP

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

Syntax

```
PURGEGROUP groupname [ .accountname ] [ ;VS=volset ]
```

Parameters

- groupname* Name of the group in the logon account to be removed. This name must contain from one to eight alphanumeric characters, beginning with an alphabetic character.
- accountname* As system manager, you can include the account name when you are logged onto an account other than the requested account.
- volset* Volume set or volume class reference which, when fully qualified, is in the form, *vcsid.groupname.acctname*. *vcsid* refers to a previously defined volume set or volume class.
- If *volset* is specified, the volume set or volume class must be mounted or the PURGEGROUP command will fail. When the *volset* parameter is specified, the group is removed from the volume set directory, and not the system directory.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have account manager (AM) capability to execute this command.

Operation

Account managers use the PURGEGROUP command to delete a group from their account. When the command is executed during a session, MPE V/E displays a verification request. Respond with "YES" or "NO" to the message:

```
GROUP groupname TO BE PURGED? (YES/NO)
```

No verification message is printed if the PURGEGROUP command is entered in a job.

If the group resides on a private volume, the command is executed only when the group's home volume set is mounted.

The order in which entries are purged is: volume set definitions, files, and finally the group. If no files in the group are in use, and the group itself is not in use, the PURGEGROUP command removes the entire group. Otherwise, only inactive files are removed. To completely purge the group in this case, re-enter the PURGEGROUP command when neither the group nor its files are in use.

Caution

Do not attempt to purge the PUB group of the SYS account. The public group of the system account, PUB.SYS, cannot be completely purged. If you specify this group in the *groupname* parameter, all non-system and inactive files are purged. This seriously impairs the functioning of the entire system.

Example

To purge a group named GROUP1, enter:

```
:PURGEGROUP GROUP1
```

```
GROUP GROUP1 TO BE PURGED? (YES/NO) YES
```

PURGEUSER

Removes a user from an account.

Syntax

```
PURGEUSER user[.accountname]
```

Parameter

<i>user</i>	Name of the user to be deleted.
<i>accountname</i>	As system manager, you can include the account name when you are logged onto an account other than the requested account.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have account manager (AM) capability to execute this command.

Operation

Account managers use the **PURGEUSER** command to delete a user from their account. You are asked to verify the command only when it is executed during a session, and not from a job. To do so, respond with “YES” or “NO” to the message:

```
USER user TO BE PURGED? (YES/NO)
```

Any attempt to purge a user currently logged onto the system will fail, and an explanatory message is displayed. This user is not affected until the next logon. An attempt to purge **MANAGER.SYS** will always fail, since this user can never be purged.

If files created by a purged user remain after the user is purged from the system, the system manager can remove them with the **PURGEACCT** command, or the account manager can eliminate them by executing **PURGEGROUP**.

Example

To remove a user named **USER1**, enter:

```
:PURGEUSER USER1  
USER USER1 TO BE PURGED? (YES/NO) YES
```

PURGEVSET

Deletes an existing volume set.

Syntax

```
PURGEVSET vsname
```

Parameters

vsname Name of an existing volume set or volume class.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have create volumes (CV) capability to execute this command.

Operation

The PURGEVSET command does not delete a volume set if it is currently in use; that is, if an explicit or implicit mount has been issued. When you enter PURGEVSET during a session, MPE V/E displays the following verification message, to which you must answer “YES” or “NO”.

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

Example

To purge the volume set called USRSET, enter:

```
:PURGEVSET USRSET  
VOLUME SET/CLASS DEFINITION USRSET TO BE PURGED? (YES/NO)  
YES
```

RECALL/=RECALL

Displays all pending console **REPLY** messages.

Syntax

RECALL

Parameters

None.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may issue it only from the console. Any user may execute the **RECALL** command, but **CTRL A**=**RECALL** may only be executed at the console.

Operation

Use the **=RECALL** command if the **RECALL** command is ineffective, or when you are executing a job or subsystem from the console.

Examples

To display all pending system console messages which require a **REPLY** response, enter:

```
:RECALL  
REPLIES PENDING:  
10:05/#J19/15/LDEV # FOR "L00576" ON TAPE1600 (NUM)?
```

If there are replies pending, the text of the request(s) is displayed on the console as shown above. If there are no replies pending, the following message appears on the console:

```
:RECALL  
NO REPLIES PENDING (CIWARN 3020)
```

REDO

Allows you to edit a command entry.

Syntax

REDO

Parameters

None.

Use

You may issue this command from a session, job, or in **BREAK**. You may not use it from a program. Pressing **BREAK** aborts execution.

Operation

The **REDO** command lets you correct errors in the last command you entered, and change it to a new, correct command without the need to completely re-enter the command.

When you enter the **REDO** command, MPE V/E displays the command to be modified and enters a mode similar to that of **EDIT/3000**. To edit the command, use the space bar on the terminal keyboard to position the cursor under the character(s) to be modified, then enter one of the following subcommands:

- | | |
|---|--|
| D | Delete. Deletes the character above the D. If D is repeated, each character above each D is deleted. |
| I | Insert. Inserts one or more characters immediately preceding the character above the I. The D and I subcommands can be used in conjunction to delete characters, then insert new characters. |
| R | Replace. Replaces the characters above the R with new characters. If one character is entered, the character above the R is replaced; if two characters are entered, two characters (the character above the R and the character to the right) are replaced. R is the default subcommand, except when the letter to be replaced is one of the other subcommands. |
| U | Undo. Cancels the effect of the previous D, I, or R subcommand. If you enter a U Return , then another U Return , the system cancels all previous subcommands for this REDO command and restores the line being corrected to its original form. |

Until you press **Return**, each of these subcommands is available to you to edit the command line.

REDO

Examples

To correct an error by inserting characters, use the I subcommand:

```
:SHOW SP
~
UNKNOWN COMMAND NAME. (CIERR 975)
:REDO
SHOW SP
  IO          ** Enter the correction and press Return. **
SHOWOUT SP   ** Press Return to execute the correction. **
NO SUCH FILE(S)
OUTFENCE = 3
```

To replace a character with another, use the R subcommand:

```
:FILE A;REC=,,G;DISC=10000,16,1;SAVE
~
:EXPECTED RECORD FORMAT OF F, V OR U.
                                         (CIERR 273)
:REDO
FILE A;REC=,,G;DISC=10000,16,1;SAVE    ** MPE V/E displays command. **
      RF                               ** Replace G with F. **
:FILE A;REC=,,F;DISC=10000,16,1;SAVE    ** Corrected command displayed. **
:                                         ** Press Return to execute. **
```

Because R is the default subcommand, you can replace the letter G by entering only the character F in the correct position:

```
:REDO
FILE A;REC=,,G;DISC=10000,16,1;SAVE    ** MPE V/E displays command. **
      F                               ** Replace G with F. **
FILE A;REC=,,F;DISC=10000,16,1;SAVE    ** Corrected command displayed. **
:                                         ** Press Return to execute. **
```

REFUSE

Disables jobs, sessions and/or data on a designated device.

Syntax

```
REFUSE [ JOBS, ] [ DATA, ] [ ldev ]
```



Parameters

JOBS	Disables the JOB (or HELLO) command from the designated device.
DATA	Disables the DATA command from the designated device.
<i>ldev</i>	The logical device number of the device for which JOB (or HELLO) and DATA commands are refused.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The **REFUSE** command prevents a device from automatically recognizing and accepting one or more of the three commands used to introduce jobs or sessions: **JOB**, **HELLO**, and **DATA**. The **JOBS** parameter in the **REFUSE** command refers to both jobs and sessions. If neither the **JOBS** or **DATA** parameter is supplied, both **JOB** (or **HELLO**) and **DATA** commands are refused. To undo the effect of the **REFUSE** command, use **ACCEPT**.

Examples

To prevent logical device #35 from recognizing the **DATA** command, enter:

```
:REFUSE DATA,35
```

To prevent both job and data recognition on logical device #35, enter:

```
:REFUSE 35
```

RELEASE

Removes all security provisions for a file with the exception of any access control definitions (ACDs). The file remains unsecured until the **SECURE** command is issued.

Syntax

RELEASE *filereference*

Parameters

filereference Actual file designator of the file whose security provisions are to be suspended, written in this format:

filename [/lockword] [.groupname] [.acctname]]

If the file has a lockword, it must be specified. If *groupname.acctname* is not specified, MPE V/E assumes the logon group and account.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **RELEASE** command removes all security provisions for a file at all levels (file, group, and account) and allows any user in the system unlimited access to the file unless an ACD for the file is defined. Other users have access to the file until its original security provisions are restored with the **SECURE** command. The file remains accessible after the job or session ends. It also remains accessible after a system failure, unless the system is restarted with a SYSDUMP tape created before the **RELEASE** command was issued for the file. You can only use this command for permanent files on disk whose labels identify you as the creator of the file. In addition, the **RELEASE** command will fail if the group's home volume set is not mounted. When the default MPE V/E security provisions are in effect, the file must be in your logon account and must belong to your logon or home group.

The **RELEASE** command removes the access attributes on a file with an associated ACD, but it does not remove ACDs themselves. Also, privileged files cannot be released.

The **RELEASE** command does not affect the file's lockword, if one was assigned. To check whether a file is secure or released, use the **LISTDIR5** utility. The prompt in the **LISTDIR5** subsystem is ">".

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

```
LISTDIR5 G.03.00 (C) HEWLETT-PACKARD CO., 1983  
TYPE 'HELP' FOR AID
```

```
>LISTF FILE1
```

```
*****
```

FILE: FILE1.MAC.TECHPUBS

FCODE: 0	FOPTIONS: STD,ASCII,FIXED	
BLK FACTOR: 4	CREATOR: **	
REC SIZE: 88(B)	LOCKWORD: **	
BLK SIZE: 176(W)	SECURITY--READ: ANY	
EXT SIZE: 4(S)	WRITE: ANY	
# REC: 4	APPEND: ANY	
# SEC: 4	LOCK: ANY	
# EXT: 1	EXECUTE: ANY	
MAX REC: 4	**SECURITY IS OFF	** FILE1 is released. **
MAX EXT: 1	COLD LOAD ID: %21604	
# LABELS: 0	CREATED: FRI, 4 MAR 1988	
MAX LABELS: 0	MODIFIED: FRI, 4 MAR 1988	
	4:17 PM	
DISC DEV #: 3	ACCESSED: WED, 16 MAR 1988	
DISC TYPE: 3	LABEL ADR: **	
DISC SUBTYPE: 0	SEC OFFSET: %2	
CLASS: DISC	FLAGS: NO ACCESSORS	
FCB VECTOR: %0	%0	

>LISTF FILE2

FILE: FILE2.MAC.TECHPUBS

FCODE: 0	FOPTIONS: STD,ASCII,FIXED	
BLK FACTOR: 29	CREATOR: **	
REC SIZE: 88(B)	LOCKWORD: **	
BLK SIZE: 1276(W)	SECURITY--READ: ANY	
EXT SIZE: 20(S)	WRITE: ANY	
# REC: 849	APPEND: ANY	
# SEC: 310	LOCK: ANY	
# EXT: 16	EXECUTE: ANY	
MAX REC: 849	**SECURITY IS ON	** FILE2 is secure. **
MAX EXT: 16	COLD LOAD ID: %21605	
# LABELS: 0	CREATED: FRI, 11 MAR 1988	
MAX LABELS: 0	MODIFIED: FRI, 11 MAR 1988	
	10:23 AM	
DISC DEV #: 3	ACCESSED: MON, 21 MAR 1988	
DISC TYPE: 3	LABEL ADR: **	
DISC SUBTYPE: 0	SEC OFFSET: %12	
CLASS: DISC	FLAGS: NO ACCESSORS	
FCB VECTOR: %0	%0	
>E		
END OF PROGRAM		

RELEASE

Examples

To release all security provisions for the file named **FILE1** in your logon group and account **MAC.TECHPUBS**, enter:

:RELEASE FILE1

If MPE V/E fails to locate the file, the following error message is displayed:

UNABLE TO ACCESS FILE1.MAC.TECHPUBS. (CIERR 356)

If you attempt to release security provisions for a file with specified ACDs (access control definitions), the following error message is displayed:

AN ACD EXISTS FOR THIS FILE. (CIWARN 402)

RELLOG

Removes a logging identifier from the system.

Syntax

RELLOG *logid*

Parameters

logid The logging identifier to be removed from the system.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have system supervisor (OP) or user logging (LG) capability in order to use this command.

Operation

The **RELLOG** command removes a logging identifier from the system by deleting it from the directory of logging identifiers. This command may be issued only by the user who created the logging identifier specified in the *logid* parameter. After **RELLOG** is issued, programs containing the removed logging identifier may not access the logging system.

Example

To remove the logging identifier **DATALOG** from the system, enter:

:RELLOG DATALOG

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005)

RENAME

Changes the identity (filename, lockword, and/or group name) of a disk file.

Syntax

```
RENAME oldfilereference, newfilereference [ ,TEMP ]
```

Parameters

<i>oldfilereference</i>	Current name of file, written in the following format: <i>filename[/lockword] [.groupname [.acctname]]</i> If you specify <i>acctname</i> , it must be that of your logon account. You must be the creator of the file to rename it.
<i>newfilereference</i>	New name of file, in the same format as <i>oldfilereference</i> . If <i>acctname</i> is specified, it must be that of your logon account. If <i>groupname</i> is specified, it must be one to which you have SAVE access. If <i>acctname</i> and/or <i>groupname</i> are omitted, the logon account and/or group are assumed.
TEMP	Indicates that the old file was, and the new file is, a temporary file. The default is that a permanent file is assumed.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **RENAME** command changes the system file identification for a permanent or temporary disk file. It can be used to change the name of a file, to move a file from one group to another (by specifying a different *groupname* in the *newfilereference* parameter), or to change the lockword. You cannot rename files that exist in other accounts. (To transfer a file from one account to another, use **FCOPY**.)

To rename a file, its group's volume set must be physically mounted or the command will fail. You must be the file's creator and have exclusive (which implies **LOCK**) access to the file. You cannot rename a file across volume sets.

Examples

Since temporary files exist only for the duration of your current job or session, their fully qualified filenames correspond to your logon group and account. To change the name of a temporary file from **OLDFILE** to **NEWFILE**, and reassign it to the group **NEWG**, enter:

```
:RENAME OLDFILE,NEWFILE.NEWG,TEMP
```

Temporary files are valid only in your logon group, and are stored according to their fully qualified file names.

To change the lockword of the permanent file **FILE2** from **LOCKA** to **LOCKB**, enter:

```
:RENAME FILE2/LOCKA,FILE2/LOCKB
```

To transfer a file from one group to another within the same account, use the **RENAME** command, simply naming the new group in the second parameter. (You must be the creator of the file to use this command.) To move the file **MYFILE** from **GROUP1** to **GROUP2**, you would enter:

```
:RENAME MYFILE.GROUP1,MYFILE.GROUP2
```

To use **RENAME** in this way, you must have **SAVE** access to the group named in the second parameter (**GROUP2** in the previous example). In addition, both groups must be in the system domain or must both reside on the same volume set.

Additional Discussion

MPE File System Reference Manual (30000-90236)

REPLY/=REPLY

Replies to pending resource requests at the console.

Syntax

REPLY *pin*, *reply*

Parameters

pin The process identification number of the message sender. As part of the message requesting the REPLY, the *pin* always appears after the second slash (/). In the following example, the *pin* is 43.

?16:15/#S25/43/LDEV# FOR "T" ON TAPE (NUM)?

reply The reply type specified in parentheses in the message, defined by one of the following:

(NUM) Reply must be a logical device number.

(Y/N) Reply must be YES (or Y), or NO (or N).

(MAX CHARS.=*nn*) Reply must be a string expression consisting of *nn* characters or less.

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** has no effect on this command. You may issue it only from the console *unless* it is distributed to specific users with the ALLOW or ASSOCIATE commands.

Operation

User programs that have requested the use of a device and are waiting for your REPLY/=REPLY remain suspended indefinitely, and cannot be aborted until the REPLY is issued. If for any reason you cannot reply as requested; for example, if the required device is nonexistent or if a special form is unavailable, then reply with 0 for a type NUM request, and with N for a type Y/N request. This returns an error code to the program and the REPLY/=REPLY command is aborted.

The reply usually takes the form of NUM or Y/N, since MAX CHARS.=*nn* is used only for labeled tapes.

If your reply is not of the type specified, if the *pin* is incorrect, or if the device you allocate is already allocated to someone else, the system displays the following error message:

NO REPLY OUTSTANDING FOR PROCESS

Examples

To reply to a message from the MPE V/E system, enter:

```
10:05/#J19/15/LDEV# FOR "NAS" OF TAPE1600 (NUM)?
:REPLY 15,7
```

or

```
=REPLY 15,7
```

To reply to a FORMS message from the MPE V/E system, enter:

```
15:46/#S93/22/FORMS: PLEASE MOUNT MAILING LABEL FORMS
?15:46/#S39/22/SP#12/LDEV# FOR #S93;OUTFILE ON LP (NUM)?
```

```
:REPLY 22,12
```

```
15:46/#S39/22/LDEV#12 FORMS ALIGNED OK (Y/N)?
```

```
** Answering NO (N) to this question causes      **
** the printing to be deferred to a much lower    **
** priority. After the forms have been properly   **
** aligned, use the ALTSPoolFILE command to      **
** change the spooling priority, in order to      **
** send the spool file to the printer.           **
```

```
:REPLY 22,NO
```

```
?15:48/#S93/22/LDEV#12 FORMS ALIGNED OK (Y/N)?
```

```
** Answering YES (Y) to this question causes      **
** the spool file to go to the printer in         **
** its assigned sequence.                         **
```

```
:REPLY 22,Y
```

When the next spool file becomes ACTIVE, you are requested to mount the appropriate special or standard forms.

To reply to a standard forms request, enter:

```
16:00/#S93/22/STANDARD FORMS
?16:00/#S93/22/LDEV # FOR #S95;L ON LP (NUM)?
```

```
:REPLY 22,12
```

REPORT

Displays accounting information for the logon account and group.

Syntax

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

Parameters

<i>groupset</i>	Specifies the accounts and groups for which information is to be listed. The permissible entries, some of which use wildcard characters, and their capability requirements, that is, account manager (AM) and/or system manager (SM), are listed below.
<i>group</i>	Reports on the specified group in the logon account. This is the default for standard users, who may specify only their logon group.
@	Reports on all groups in the logon account. This is the default for account managers, but may be executed by users with AM or SM capability.
<i>group.acct</i>	Reports on the specified group in the specified account. This requires SM capability.
@.acct	Reports on all groups in the specified account. This requires AM capability (if it is the logon account) or SM capability for any account.
@.@	Reports on all groups in all accounts. This is the default for system managers and requires SM capability.
<i>group.@</i>	Reports on a specified group in any account. It also lists all account totals alphabetically, even those where the specified group does not exist. This requires SM capability.

The characters @, #, and ? can be used as wildcard characters, but count toward the eight-character limit. These wildcard characters have the following meanings.

@	Specifies zero or more alphanumeric characters and denotes all members of the set.
#	Specifies one numeric character.
?	Specifies one alphanumeric character.

The characters can be used as follows.

n@	Report on all groups starting with the character "n".
@n	Report on all groups ending with the character "n".
n@x	Report on all groups starting with the character "n" and ending with the character "x".

REPORT

n## ... #	Report on all groups starting with the character “n” followed by up to seven digits.
?n0	Report on all groups whose second character is “n”.
n?	Report on all two-character groups starting with the character “n”.
?n	Report on all two-character groups ending with the character “n”.

These characters, when placed appropriately in the *groupset* parameter, may also be used to report on accounts.

listfile Actual file designator of the output file to which information is to be written. The default is \$STDLIST, but output may be redirected with a file equation as follows:

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

volset Informs MPE V/E to report accounting information for the specified volume set.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** aborts execution. You must have account manager (AM) or system manager (SM) capability to execute this command.

Operation

The **REPORT** command displays the total resource usage logged against accounts and groups, and the limits on those resources. For standard users, data is displayed for their group only; an account manager may specify all groups in his or her account; the system manager may specify any or all groups in any or all accounts.

The information includes usage counts and limits for permanent file space (in sectors), CPU time (in seconds), and session connect time (in minutes). The file space usage count reflects the number of sectors used at the time the **REPORT** command was issued. However, CPU time and connect time usage appear as they were immediately before the beginning of the current job.

REPORT

Example

To obtain accounting information for your group, enter the **REPORT** command. Accounting information is displayed in a format similar to that of the **SOPRM** account shown below:

:REPORT @.SOPRM

ACCOUNT	FILESPEC-SECTORS		CPU-SECONDS		CONNECT-MINUTES	
/GROUP	COUNT	LIMIT	COUNT	LIMIT	COUNT	LIMIT
SOPRM	99004	**	99057	**	88407	**
/GLOSSARY	1068	**	542	**	656	**
/PUB	182	**	123	**	1155	**
/SECT1	180	**	85	**	429	**
/SECT10	11779	**	25271	**	9716	**
/SECT2	390	**	4123	**	5302	**
/SECT3	10675	**	8176	**	13116	**
/SECT4	2372	**	225	**	294	**
/SECT5	46579	**	27218	**	25744	**
/SECT6	6008	**	9324	**	6638	**
/SECT7	4748	**	8303	**	13263	**
/SECT8	1957	**	6348	**	3997	**
/SECT9	3195	**	4570	**	4213	**

RESET

Cancels file equations.

Syntax

$$\text{RESET} \left\{ \begin{array}{l} \text{formaldesignator} \\ @ \end{array} \right\}$$

Parameters

formaldesignator A formal file designator name in the form
 file[.group[.account]][:nodespec]
 for which a **FILE** command was previously issued. The *nodespec*
 portion may be an environment identifier indicating the location of
 the file, or it may be **\$BACK**. Specifying **\$BACK** means that the file
 resides one “hop” back toward your local system (which may be the
 local system itself).

@ Signifies all formal file designators specified in all **FILE** commands
 previously issued in this session or job.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **RESET** command resets a formal file designator to its original meaning, cancelling any **FILE** command that has been issued for this formal file designator earlier in the current session or job.

Note



The *nodespec* parameter is not part of the HP 3000 fundamental operating system. The NS/3000 AdvanceNet subsystem must be purchased separately. The *nodespec* parameter is optional: if you do not have NS/3000 AdvanceNet, omitting the *nodespec* parameter makes no difference to the performance of the **RESET** command. However, specifying *nodespec* on a system that does not have NS/3000 produces an error.

Example

To cancel the effects of a previous **FILE** command that specified characteristics for a file named **ALPHA**, enter:

:RESET ALPHA

Additional Discussion

Refer to the **FILE** command in this chapter.

RESETACCT

Zeroes the running counts of CPU time or connect time accumulated by an account and by all groups within that account.

Syntax

```
RESETACCT [ [ @ ] [ , [ CPU ] ] ] [ acct ] [ , [ CONNECT ] ] ]
```

Parameters

@	Specifies that the counters for all accounts, and all groups within the accounts, are to be reset. This is the default.
<i>acct</i>	Specifies the name of a particular account, and all groups within the account, for which the CPU and connect time are to be reset. The default is @.
CPU	Specifies that only the CPU usage counter is to be reset. The default is that both the CPU and connect time counters are reset.
CONNECT	Specifies that only the connect time usage counter is to be reset. The default is that both the CPU and connect time counters are reset.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You must have system manager (SM) capability to execute this command.

Operation

The **RESETACCT** command zeroes the running counts of CPU or connect time accumulated by an account and by all groups within that account. If you omit all parameters when you execute **RESETACCT**, all counters (except file space) for all groups in all accounts are reset.

Example

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

```
:RESETACCT @,CPU
```

RESETDUMP

Disables the stack dump facility.

Syntax

RESETDUMP

Parameters

None.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command.

Operation

The **RESETDUMP** command disables the MPE V/E stack dump facility (enabled by the **SETDUMP** command). For more information, refer to the *MPE Debug/Stack Dump Reference Manual* (30000-90012). When entered in **BREAK**, **RESETDUMP** does not modify the state of processes already created. If the stack dump facility is not enabled, the **RESETDUMP** command has no effect.

Example

To disable the stack dump facility, enter:

:RESETDUMP

Additional Discussion

MPE Debug/Stack Dump Reference Manual (30000-90012)

RESTORE

Returns files that have been stored on magnetic tape or serial disk to the system.

Syntax

```
RESTORE[ restorefile ] [ ; { fileset [ -fileset ] [ , fileset [ -fileset ] ] [ , ... ] } ]  
[ ; DEV=device ] [ ; SHOW=[ showparm [ , showparm [ , ... ] ] ] [ ; FILES=maxfiles ]  
[ ; { LOCAL  
  GROUP=groupname  
  ACC[oun]T=acctname } [ ; ... ] ] [ ; CREATE [ = { GROUP  
  ACCT  
  CREATOR } [ , ... ] ] ]  
[ ; CREATOR[=username] ] [ ; { KEEP  
  NOKEEP } ] [ ; { OLDDATE  
  NEWDATE } ] [ ; ONERR[or]= { QUIT  
  SKIP } ]  
[ RESTORESET=tapelistspec ] [ ; STARTHERE ] [ ; COPYACD ]
```

Parameters

restorefile The name of the input device on which the files to be restored exist. If you have previously referenced the device in a **FILE** command, you can backreference it with an asterisk (*). If you do not specify an input device, **RESTORE** uses the logon user name and the default device, **TAPE**. To use multiple input devices, use the **RESTORESET** parameter.

restorefile If you choose not to specify *restorefile*, a default tape name is used. The default tape name is the same as the user's logon identification and MPE V/E designates **TAPE** as the device.

fileset Specifies the set or sets of files to be restored. This parameter has the form:

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

You can use the characters **@**, **#**, and **?** as wildcard characters to refer to a group of files. The wildcard characters count toward the eight-character limit for the user, group, account, and file names. The wildcard characters have the following meanings:

- @** Represents zero or more alphanumeric characters. Used alone, **@** denotes all members of the set.
- #** Represents one number.
- ?** Represents one alphanumeric character.

You may use the wildcard characters as shown in the following examples:

- n** Stores all items beginning with the character “n”.
- n** Stores all items ending with the character “n”.
- n**x Stores all items starting with the character “n” and ending with the character “x”.
- n#####** Stores all items starting with the character “n” followed by seven digits.
- ?n** Stores all items whose second character is “n”.
- n?** Stores all two-character items beginning with an “n”.
- ?n** Stores all two-character items ending with an “n”.

If you do not specify an account or group, MPE V/E assumes the logon account or group. The *fileset* default for standard users is **n**, which restores all files in the logon group.

You can use more than one *fileset* (and an associated *-fileset*) with a single **RESTORE** command. The maximum number of file sets is limited to all files in up to 10 different accounts, all files belonging to 15 different groups within the same account, or 20 different files within the same group and account.

-fileset

Name of a file set or subset to be excluded. You can have only one excluded subset for each file set, and you cannot place excluded file sets within other excluded file sets. A minus sign (-) must precede this parameter.

!indirectfile

An ASCII file which contains the desired parameters of the **RESTORE** command. Each line of the indirect file may contain file sets and keywords. Use an indirect file when you need to restore the same set of files regularly or when your **RESTORE** command exceeds the CI's 268-character limit. An exclamation point (!) must precede this parameter.

device

Device class (*devclass*) or logical device number (*ldv*) of the device on which the file is to reside. If you specify a device class, the **RESTORE** command allocates the file to any volume of the volume set within that class. A logical device number designates a specific device. When you specify a logical device number, **RESTORE** allocates the file to that device only if one of the volumes in the volume set currently occupies that device.



RESTORE

By default, **RESTORE** attempts to restore a file on the same device class as specified in the file's label. If it cannot, **RESTORE** tries to restore the file on a logical device compatible with the type and subtype specified in the file label. If a second attempt fails, **RESTORE** tries to restore the file to the device class **DISC**. If all three attempts fail, MPE V/E does not restore the file.

SHOW

Request to list names of restored files. By default, **RESTORE** lists the total number of all files restored and not restored and the reason a file was not restored. Unless you use a **FILE** command to redirect the listing, **RESTORE** sends it to **\$STDLIST** (formal file designator **SYSLIST**).

showparm

May be **SHORT**, **LONG**, **DATES**, **SECURITY**, or **OFFLINE**. You cannot use both **LONG** and **SHORT**, but you can combine all other options, separating them by commas. If you do not select **SHORT** or **LONG**, **RESTORE** uses **SHORT** if **\$STDLIST** is less than 99 characters and **LONG** otherwise.

- | | |
|-----------------|---|
| SHORT | Displays the file name, group name, account name, logical device number, reel number, and file size (in sectors) of all restored files. |
| LONG | Displays the same information as SHORT and the file record size, file type, end-of-file, file record limit, blocking factor, extents allocated and maximum extents of all restored files. |
| DATES | Displays the creation date, last access date, and last modification date of all files restored. |
| SECURITY | Displays the creator and file access matrix of all files restored. |
| OFFLINE | Sends file system listings to the system line printer as well as to your terminal. You can redirect the output from the system line printer to another output device with a FILE command redefining the formal file designator as OFFLINE . |

maxfiles

Maximum number of files that may be restored. If you are restoring from a large tape or disk, such as a **SYSDUMP** tape, you must include this parameter or the file(s) will not be restored. The value of *maxfiles* should be at least as large as the maximum number of files on the device from which you are restoring. The default is 4,000. This option may not be included in an indirect file; instead, it should be specified in the **RESTORE** command.

LOCAL , <i>groupname</i> , or <i>acctname</i>	Restores files from different systems or directory structures. LOCAL restores all files to the user's creator identification, group, and account. To restore a file with the LOCAL option, you must have READ access to the tape file. Check your capabilities against the account, group and file securities of the tape file. You must have SM capability to restore a file from a nonexistent account or across an existing account. You need AM , SM , or OP capability to restore a file from a nonexistent group of your logon account. If you specify a <i>groupname</i> , MPE V/E changes file group names to the name you specify.
CREATE= GROUP , ACCOUNT , OR CREATOR	Builds groups, accounts, or creators that do not currently exist in the directory. When it creates a group, account, or creator, RESTORE sends a message to SYSLIST (default \$STDLIST). GROUP creates the group specified in the file label and requires system manager (SM) or account manager (AM) capability. ACCOUNT (or its short form ACCT) creates the account specified in the file label and requires system manager (SM) capability. CREATOR creates the creator described in the file label and requires account manager (AM) capability. By default, RESTORE creates directory components based upon your capabilities.
CREATOR= <i>username</i>	Changes all files' creator identifications to <i>username</i> . If <i>username</i> does not exist in the directory, the file is not restored unless CREATE is specified. Do not use the "equals" sign if <i>username</i> is unspecified. If you do not specify CREATOR= <i>username</i> , RESTORE uses the creator named in the file label when that creator exists in the directory. If the creator does not exist, it does not restore the file unless CREATE is specified.
KEEP or NOKEEP	You use these two parameters to control whether or not RESTORE replaces files that already exist on disk. KEEP instructs RESTORE to keep disk files and not to restore the tape file. (If RESTORE cannot restore the tape file, it keeps the disk file.) The default is NOKEEP .
NEWDATE or OLDDATE	MPE V/E stores four dates in each file label: the creation date, modification date, last access date, and the allocation date and time. NEWDATE changes all dates and times to the date and time that you restored the file. OLDDATE retains the original modification and last access dates, but changes the creation and allocation dates to the date and time you restored the file.

RESTORE

QUIT or SKIP

Directs **RESTORE** to quit or continue when it encounters a tape error. When you specify **SKIP**, **RESTORE** skips to the next file after the tape error and begins reading files from the tape. It does not restore the files that it was unable to read. When you use **QUIT**, **RESTORE** terminates upon encountering a tape error. The default and only option for labeled tapes is **QUIT**. The default for unlabeled tapes is **SKIP**.

STARTHERE

RESTORE usually requires you to mount the first tape volume containing any of the requested files. If a tape containing needed files is not available, you can mount the first available tape and specify **STARTHERE** as an option in your **RESTORE** command. This causes requested files on prior volumes to be listed but not restored. Restoration begins with the first qualifying file on the mounted tape.

RESTORESET

Specifies input from multiple devices. All devices must have the same device type, device subtype, density, and record size. You can only use **RESTORESET** if you purchase TurboSTORE (see note below).

Note



MPE V/E has two versions of **STORE** and **RESTORE**: the standard **STORE/RESTORE** and TurboSTORE. Standard **STORE/RESTORE** is part of the fundamental operating system (FOS). TurboSTORE is a separate product. When installed on your system, TurboSTORE adds additional capabilities to the **STORE** and **RESTORE** commands. This manual describes both versions of **STORE** and **RESTORE**.

tapelistspec

Describes input devices for the **RESTORESET** parameter. It has the following syntax:

(tapedevice) [, *tapedevice*] [...] [, *tapedevice*]

tapedevice specifies an output device. It must backreference a **FILE** command.

COPYACD

Specifies that access control definitions (ACDs) are to be copied. If you specify the **COPYACD** option, any ACDs associated with the file are restored with the file. If no ACDs are stored with the file, the file is restored and an asterisk is printed after the filename. The default is that ACDs associated with a file are not copied.

If you use wildcards to define file names, the system tries to restore each file with its associated ACD. If an ACD associated with a file is not restored, an appropriate message is displayed and processing continues. If an ACD exists on a file and **COPYACD** is not specified, then the file is restored without the ACD.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** aborts execution. You must have system manager (SM), system supervisor (OP), or privileged mode (PM) capability to restore privileged files.

Operation

The **RESTORE** command transfers files stored on tape via the **STORE** or **SYSDUMP** program to disk. Before you enter a **RESTORE** command, identify the devices from which you intend to restore files with a **FILE** command. **RESTORE** sends messages to the system console requesting the system operator to mount the tape or serial disk device and to allocate the device.

RESTORE sends the output it generates to a file whose formal designator is **SYSLIST**. If **RESTORE** encounters a directory error while updating the directory, updating the disk free space tables, or writing the data to the disk file, it reports the error to **SYSLIST** (defaults to **\$STDLIST**) and continues. **RESTORE** does not restore any file belonging to a group whose home volume set has not been mounted.

The **RESTORE** command resides in a program file (**STORE.PUB.SYS**), rather than a set of procedures with no global memory. Both the **STORE** and **RESTORE** commands initiate this program. Use **ALLOCATE STORE.PUB.SYS** to speed up the loading process and reduce the use of system resources. If you are restoring files from a large tape or disk, such as a **SYSDUMP** tape, you must include the *maxfiles* parameter or the file(s) will not be restored. The value of *maxfiles* should be at least as large as the maximum number of files on the device from which you are restoring files.

If you do not specify a *restorefile*, then **RESTORE** uses a default tape name. The default tape name is equivalent to the user's logon identification and the device is **TAPE**. For example, if you are logged onto **TOM.MGR**, the tape request is for **TOM**. You cannot use a file equation to change this default. Your capabilities determine which files you can restore. If you have **SM** or **OP** capability, you can restore any file, assuming the account and group to which the file belongs, and the user who created the file, are defined in the system. If you have **AM** capability, you can restore any file in your account. To restore files with negative file codes, you need privileged mode capability. If you have standard user capability, you can restore files in your logon account.

MPE V/E allows you to build groups, accounts, and creators which do not currently exist in the directory. This way, you can restore files to your system without first defining the account, group, and user with the **NEWACCT**, **NEWGROUP**, and **NEWUSER** commands.

The system manager and system supervisor can restore lockword-protected files without specifying the lockword. Users without **SM** or **OP** capability must always supply the lockword.

You can issue the **RESTORE** command with the **RUN STORE.PUB.SYS** command. Use the **INFO=" "** parameter to specify the options, file sets, and keywords; in this case, the **RESTORE** command executes with no further interaction. If you do not include **INFO=" "**, the "**<--**" prompt appears. Acceptable responses are: a complete **STORE** command, a complete **RESTORE** command, an **MPE V/E** command preceded by a colon (:), or **EXIT**.

RESTORE

When you use the RUN command, the system-defined JCW CIERROR does not change even if RESTORE aborts. When you issue a RESTORE command from the command interpreter (for example, RESTORE *T;@) and the RESTORE command aborts for any reason, the system-defined JCW CIERROR changes to the value "1091". Check this value with the SHOWJCW command after RESTORE aborts.

Note



RESTORE determines whether sufficient disk space remains to restore a file that already exists on the disk. If sufficient space remains, RESTORE writes a new copy of the file to the disk before purging the old copy of the file. The old copy is purged only if the RESTORE operation is successful.

If insufficient space is available, RESTORE first purges the old copy of the file and then writes a new copy to the disk. If the RESTORE operation fails in this circumstance, you receive a message on \$STDLIST informing you that there is no copy of the file on the disk.

Examples

To restore all files belonging to your logon group from the *restorefile* named T, enter:

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

In response, the system operator receives a request to mount the tape identified as T. If a disk file on T already exists in the system, it will not be restored because the KEEP parameter was specified. To restore a file ABC without specifying a *restorefile*, you do not need to use a file equation. For example:

```
:RESTORE;ABC.PUB.SYS;SHOW
STORE/RESTORE, VERSION 2 (C) 1981 HEWLETT-PACKARD CO.
WED, AUG 2, 1989 11:56 AM

WILL RESTORE      1 FILES;  NUMBER OF FILES ON TAPE =   87

FILENAME.GROUP    .ACCOUNT LDN  ADDRESS REEL  SECTORS CODE

ABC      .PUB      .SYS      2%00100217    2      4

FILES RESTORED:      1
FILE NOT RESTORED:   0
:
```

When you restore files without specifying a *files*et, you see a warning message telling you that all files, based on your capabilities, will be restored:

```
:RESTORE
STORE/RESTORE, VERSION 2   (C) 1981  HEWLETT-PACKARD CO.
FRI, APR 12, 1985, 11:39 AM

RESTORE
~
****WARNING:  AN 'EMPTY' FILESET WAS FOUND ... BASED ON YOUR
CAPABILITIES, '0.0.0' WILL BE ASSUMED**** (S/R 6229)
```

To have the list of restored files printed on a line printer, enter:

```
:FILE SYSLIST;DEV=LP
:FILE T;DEV=TAPE
:RESTORE *T;0;SHOW
```

To restore the file FILEA.GROUPA.ACCOUNTA when the creator, USERA, does not exist on the system, use one of the methods shown below:

```
:RESTORE *TAPEFILE; FILEA.GROUPA.ACCOUNTA; CREATOR=USERB
```

(Changes the creator of FILEA to USERB. USERB must exist on the system.)

Or:

```
:RESTORE *TAPEFILE; FILEA.GROUPA.ACCOUNTA; CREATE=CREATOR
```

(Creates USERA on the system.)

Or:

```
:RESTORE *TAPEFILE; FILEA.GROUPA.ACCOUNTA; CREATE
```

(Creates USERA on the system.)

To restore only a subset of the files

et, enter:

```
:RESTORE *T;0.0.0 -0.PUB.SYS
```

(Restores all files except those in the PUB group of the SYS account.)

To restore files using the RUN command, enter:

```
:RUN STORE.PUB.SYS; INFO="RESTORE *T;0.0.0;SHOW"
```

To restore files from two sequential devices, issue two FILE commands to name the devices. Then use those file names in your RESTORE command as shown below:

```
:FILE TAPE1;DEV=TAPE
:FILE TAPE2;DEV=TAPE
:RESTORE 0.0.0;RESTORESET=(TAPE1,TAPE2)
```

Additional Discussion

MPE V System Operation and Resource Management Reference Manual (32033-90005)

RESUME

Resumes execution of a suspended operation.

Syntax

RESUME

Parameters

None.

Use

You may issue this command only while in **BREAK**. You may not use it from a session (other than while in **BREAK** mode), job, or program. Pressing **BREAK** has no effect on this command.

Operation

After you have suspended a program or MPE V/E command by pressing **BREAK** or by using the **-CAUSEBREAK** intrinsic, the **RESUME** command resumes execution of the operation at the point where the execution was suspended. The **RESUME** command is legitimate only during a **BREAK**. Many MPE V/E commands are aborted rather than suspended by a **BREAK**, and thus cannot be resumed. (Refer to Table 1-1 for a list of nonbreakable commands.)

If, instead of **RESUME**, you enter another program command (such as **EDITOR**, **FORTRAN** or **RUN**) or one of the nonprogram commands (**HELLO**, **BYE**, **JOB**, or **DATA**), the command interpreter prints the following message on your terminal:

ABORT? (YES/NO)

If you respond "YES" to the **ABORT?** message, the command interpreter aborts the current program and executes the command.

If you respond "NO" to the **ABORT?** message, the command interpreter prints this message :

COMMAND NOT ALLOWED IN BREAK

It then prompts you for another command. If you now enter **RESUME** at the colon (:) prompt, the suspended program continues at the point where it was interrupted. If you originally logged on using () **COMMAND LOGON** with a program command inside the parentheses, then MPE V/E aborts the command and logs you off immediately when you respond "YES" to the "ABORT?" message.

Example

To continue a suspended program at the point of interruption, enter:

:RESUME

READ PENDING

Return

RESUMEJOB

Resumes a suspended job.

Syntax

```
RESUMEJOB #Jnnn
```

Parameters

#Jnnn A job number.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only from the console *unless* it is distributed to specific users with the **ALLOW** command, or if **JOBSECURITY** is set at **LOW**.

Operation

The system operator uses the **RESUMEJOB** command to resume processing a job suspended with the **BREAKJOB** command. The job continues executing from the point at which it was suspended; no message is issued.

Example

To resume the processing of job #668, enter:

```
:RESUMEJOB #J68
```

RESUMELOG

Resumes system logging following suspension caused by an error.

Syntax

RESUMELOG

Parameters

None.

Use

You may issue this command from a session, job, program, or in **BREAK**. You require system supervisor (OP) capability to use it. You may execute it only from the console *unless* it is distributed to users with the **ALLOW** command.

Operation

There are two types of system logging errors; recoverable errors and irrecoverable errors. The type of error is identified by the error number.

If a recoverable error occurs, you must invoke **RESUMELOG** after the error is corrected and the system configured to allow other users to log back on to the system.

An irrecoverable error is a catastrophic error which is either a physical input/output error or a unit failure. If one occurs, **RESUMELOG** cannot resume the logging facility. The system must be shut down and then restarted.

When the system operator directs MPE V/E to resume logging with **RESUMELOG**, a special log record is displayed that denotes the number of log events and corresponding records that were not recorded while logging was suspended; the total number of unrecorded job initiation records; and the total number of unrecorded job/session termination records.

Examples

Assume the system is online and running with logging enabled. If a recoverable error occurs, the following error message is sent to the system console:

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

After the error is corrected, enter the **RESUMELOG** command. A confirmation message then appears at the system console, as follows:

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

Additional Discussion

Refer to the discussion of system logging in “MPE Logging Facilities” in the *MPE V System Operation and Resource Reference Manual* (32033-90005).

RESUMESPOOL

Resumes suspended spooler output to a spooled device.

Syntax

```
RESUMESPOOL ldev;BACK  $\left[ \begin{array}{l} nnn \text{ FILES} \\ nnn \text{ PAGES} \end{array} \right]$ 
```

```
RESUMESPOOL ldev;FORWARD  $\left[ \begin{array}{l} nnn \text{ FILES} \\ nnn \text{ PAGES} \end{array} \right]$ 
```

```
RESUMESPOOL ldev;BEGINNING
```

Parameters

<i>ldev</i>	The logical device number of a spooled device.
BACK	Instructs the spooler to backspace <i>nnn</i> files or <i>nnn</i> pages and resume printing at that point. (Refer to “OPERATION”.)
FORWARD	Instructs the spooler to space forward <i>nnn</i> files or <i>nnn</i> pages and resume printing at that point. (Refer to “OPERATION”.)
BEGINNING	Instructs the spooler to resume printing at the beginning of the file which had been previously suspended.
<i>nnn</i>	The number of files or pages you wish the spooler to backspace or space forward when printing resumes. (Must be an integer between 1 and 256, inclusive.)
FILES or PAGES	Informs the spooler process which unit of measure to use when printing resumes. For the purposes of this command, FILE is defined as the text appearing between FOPEN intrinsic statements within the spool file. (Refer to “OPERATION”.) PAGE is the literal page (usually 60 lines or skip to channel 1), as output by the spooler to the printer.

Note



The use of the **FILES** parameter is not allowed on the HP 2680A page printer or on an HP 2608S CIPER-protocol printer.

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** has no effect on this command. You may execute it only at the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

RESUMESPOOL

Operation

The **RESUMESPOOL** command lets you resume printing suspended output from a spool file or files. If you specify only the *ldev* parameter, the printer resumes printing at the beginning of the highest priority spool file. Otherwise, the printer resumes printing the previously active spool file.

Always overestimate the number of files or pages you need when using the **BACK** parameter or underestimate the number when using the **FORWARD** parameter. This is the only way to ensure that you get all the output you need, since partial pages and header pages affect the page count. However, if you instruct the spooler to go back further than the beginning of the file, MPE V/E displays an error message on the system console and printing resumes at the beginning of the file. Similarly, an error message is generated if you instruct the spooler to advance forward beyond the point where files exist. In this case, printing does not resume until a new command is issued.

By using the **SPOOK** utility with mode control on, you can determine where each **FOPEN** intrinsic occurs within a spool file. This is useful, for example, when you are compiling, preparing, and running large programs, and printing the entire output is unnecessary. For information about the **SPOOK** utility, refer to the *MPE V System Utilities Reference Manual* (32033-90008).

Examples

To resume output to logical device #6 at the beginning of the file, enter:

```
:RESUMESPOOL 6;BEGINNING
```

To resume output to logical device #6, and reprint the last two pages, enter:

```
:RESUMESPOOL 6;BACK 2 PAGES
```

To resume output to logical device #6, and print the highest priority spool file, enter:

```
:RESUMESPOOL 6
```

RPG

Compiles an RPG program. RPG is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax



```
RPG [ textfile ] [ , [ uslfile ] [ , [ listfile ] [ , [ masterfile ] [ , newfile ] ] ] ] ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is RPGTEXT. The default is \$STDIN.
<i>uslfile</i>	Actual file designator of the user subprogram library (USL) file to which the object program is written. This can be any binary input file with a file code of USL or 1024. Its formal file designator is RPGUSL. If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS. If entered, this parameter refers to a file created in one of four ways: ■ By saving, with the SAVE command. The default is the USL file created during a previous compilation. ■ By building the USL with the Segmenter-BUILDUSL command. (Refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011)). ■ By creating a new USL file with the BUILD command and specifying a file code of USL or already fixed in file 1024. ■ By specifying a nonexistent <i>uslfile</i> parameter, thereby creating a permanent file of the correct size and type.
<i>listfile</i>	Actual file designator of the file on which the program listing is written. This can be any ASCII output file. The formal file designator is RPGLIST. The default is \$STDLIST.
<i>masterfile</i>	Actual file designator of the master file to be merged against <i>textfile</i> to produce a composite source. This can be any ASCII input file. The formal file designator is RPGMAST. The default is that the master file is not read, and input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified. If two files being merged have identical line numbers, the lines from <i>textfile</i> or from \$STDIN overwrite those in <i>masterfile</i> .
<i>newfile</i>	Actual file designator of the file produced by merging <i>textfile</i> and the <i>masterfile</i> . This can be any ASCII output file. The formal file designator is RPGNEW. The default is that no file is written.

Note



The formal file designators used in this command (RPGTEXT, RPGUSL, RPGLIST, RPGMAST, and RPGNEW) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on "Implicit FILE Commands For Subsystems" under the FILE command.

RPG

Use

You may issue this command from a session or job. You may not issue it in **BREAK** or from a program. Pressing **BREAK** suspends execution.

Operation

The RPG command compiles an RPG program onto a user subprogram library (USL) file on disk. If you do not specify *textfile*, MPE V/E expects input from your standard input device. If you create the USL file before compiling the source code, you must assign it a file code of USL or 1024.

Examples

The following example compiles an RPG program entered from your standard input device, storing the object code in the default USL file **\$OLDPASS**. The listing is sent to the standard list device.

```
:RPG
```

The next example compiles an RPG program contained in the disk file **SOURCE**. The object code is stored in the USL file **OBJECT**, which is a permanent disk file created with the **BUILD** command. The program listing is sent to the disk file **LISTFL**.

```
:BUILD OBJECT;CODE=USL  
:RPG SOURCE,OBJECT,LISTFL
```

To compile an RPG program, storing the object code in the USL file **OBJECT** (created during the compilation process) enter:

```
:RPG SOURCE,OBJECT,LISTFL
```

Additional Discussion

RPG/3000 Compiler Reference Manual (32104-90001)
MPE Segmenter Reference Manual (30000-90011)

RPGGO

Compiles, prepares, and executes an RPG program. RPG is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
RPGGO [ textfile ] [ , [ listfile ] [ , [ masterfile ] [ , newfile ] ] ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which source program is read. This can be any ASCII input file. The formal file designator is RPGTEXT . The default is \$STDIN .
<i>listfile</i>	Actual file designator of the file on which the program listing is written. This can be any ASCII output file. The formal file designator is RPGLIST . The default is \$STDLIST .
<i>masterfile</i>	Actual file designator of a file which is merged against <i>textfile</i> to produce a composite source. This can be any ASCII input file. The formal file designator is RPGMAST . The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified. If two files being merged have identical line numbers, the lines from <i>textfile</i> or from \$STDIN overwrite those in <i>masterfile</i> .
<i>newfile</i>	Actual file designator for the file produced by merging <i>textfile</i> and the <i>masterfile</i> . This can be any ASCII output file. The formal file designator is RPGNEW . The default is that no file is written.

Note



The formal file designators used in this command (**RPGTEXT**, **RPGLIST**, **RPGMAST**, and **RPGNEW**) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the **FILE** command.

Use

You may issue this command from a session or a job. You may not issue it in **BREAK** or from a program. Pressing **BREAK** suspends execution.

Operation

The **RPGGO** command compiles, prepares, and executes an RPG program. If you do not specify *textfile*, MPE V/E expects you to enter the source code from your standard input device.

The USL file created during compilation is a system-defined temporary file **\$OLDPASS**, which is passed directly to the Segmenter. It cannot be accessed, since the Segmenter also uses **\$OLDPASS** to store the prepared program segments and overwrites the USL file of the same name.

RPGGO

Examples

To compile, prepare, and execute an RPG program entered from your standard input device, sending the program listing to your standard list device, enter:

:RPGGO

To compile, prepare, and execute an RPG program **READ** from the disk file **SOURCE**, sending the program listing to the disk file **LISTFL**, enter:

:RPGGO SOURCE,LISTFL

Additional Discussion

RPG/3000 Compiler Reference Manual (32104-90001)
MPE Segmenter Reference Manual (30000-90011)

RPGPREP

Compiles and prepares an RPG program. RPG is not part of the HP 3000 fundamental operating system and must be purchased separately.

Syntax

```
RPGPREP [ textfile ] [ , [ progfile ] [ , [ listfile ] [ , masterfile ] [ , newfile ] ] ]
```

Parameters

<i>textfile</i>	Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is RPGTEXT . The default is \$STDIN .
<i>progfile</i>	Actual file designator of the program to which the prepared program segments are written. You get the best results by omitting the <i>progfile</i> parameter. When you omit <i>progfile</i> , the Segmenter creates the program file, which resides in the temporary file domain as \$OLDPASS . If you do create your own program file, however, you must do so in one of two ways: ■ By using the BUILD command, and specifying a file code of 1029 or PROG, and a <i>numextents</i> value of 1. This file is then used by the PREP command. ■ By specifying a nonexistent file in the <i>progfile</i> parameter, in which case a job or session file of the correct size and type is created. The default is that \$NEWPASS is assigned.
<i>listfile</i>	Actual file designator of the file on which the program listing is written. This can be any ASCII output file. The formal file designator is RPGLIST . The default is \$STDLIST .
<i>masterfile</i>	Actual file designator of the master file that is merged against <i>textfile</i> to produce a composite sourcefile. This can be any ASCII input file. The formal file designator is RPGMAST . The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified. If two files being merged have identical line numbers, the lines from <i>textfile</i> or from \$STDIN overwrite those in <i>masterfile</i> .
<i>newfile</i>	Actual file designator of the file produced by merging <i>textfile</i> and the <i>masterfile</i> . This can be any ASCII output file. The formal file designator is RPGNEW . The default is that no file is written.

Note



The formal file designators used in the command (**RPGTEXT**, **RPGLIST**, **RPGMAST**, and **RPGNEW**) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the **FILE** command.

RPGPREP

Use

You may issue this command from a session or a job. You may not issue it in **BREAK** or from a program. Pressing **BREAK** suspends execution.

Operation

The RPGPREP command compiles and prepares an RPG program to a program file on disk. If you do not specify *textfile*, MPE V/E expects the source program to be entered from your standard input device. The USL file \$OLDPASS, created during compilation, is a system-defined temporary file passed directly to the Segmenter. You can access it only if you do not use the \$NEWPASS default for *progfile*. This is because the Segmenter also uses \$OLDPASS to store the prepared program segments, overwriting any existing temporary files of that name.

Examples

The compile and prepare an RPG program entered from your standard input device, sending the listing to your standard list device, enter:

```
:RPGPREP
```

The USL file created during compilation is a temporary file passed directly to the Segmenter. You can access it under the name \$OLDPASS only if the prepared program segments are not also stored in \$OLDPASS (which overwrites the USL file). Therefore, to save the compiled USL and the prepared program file, specify a non-existent file for *progfile* in the RPGPREP command line and save the USL file \$OLDPASS under another name. In the following example, the prepared program is saved as COMFL, and the USL file is renamed (and saved) to NUSL:

```
:RPGPREP,COMFL  
:SAVE $OLDPASS,NUSL
```

Unless you have specifically created a permanent file to store the prepared program, the program file COMFL is stored in the temporary file domain. To save it as a permanent file, use the SAVE command:

```
:SAVE COMFL
```

Using the BUILD command, you can create your own program file in the permanent file domain. When you do so, be sure to specify a file code of "PROG" or "1029" and a *numextents* parameter value of 1. Such a file is created in the next example. It is then used by the PREP command.

```
:BUILD PROGFL;CODE=PROG;DISC=,1  
:RPGPREP,PROGFL
```

To send the program listing to a device other than the default standard list device, use the **FILE** command. In this example, the file equation assigns the filename **LINEA** to device class **LP** (your line printer). **LINEA** is then backreferenced in the **RPGPREP** command line:

```
:FILE LINEA;DEV=LP  
:RPGPREP,EDTDISC,COMFL,*LINEA
```

Additional Discussion

RPG/3000 Compiler Reference Manual (32104-90001)

MPE Segmenter Reference Manual (30000-90011)

RUN

Executes a prepared program.

Syntax

$$\begin{aligned} \text{RUN } \textit{progfile} [, \textit{entrypoint}] \quad [; \text{NOPRIV}] \quad & \left[; \text{LIB} = \left\{ \begin{array}{c} \text{G} \\ \text{P} \\ \text{S} \end{array} \right\} \right] \quad [; \text{LMAP}] \quad [; \text{NOCB}] \quad [; \text{DEBUG}] \\ [; \text{INFO} = \textit{quotestring}] \quad [; \text{MAXDATA} = \textit{segsize}] \quad & \left[; \text{STDIN} = \left\{ \begin{array}{c} * \textit{formaldesig} \\ \textit{fileref} \\ \$\text{NULL} \end{array} \right\} \right] \\ [; \text{PARM} = \textit{parameternum}] \quad & \left[; \text{STDLIST} = \left\{ \begin{array}{c} * \textit{formaldesig} \\ \textit{fileref} [, \text{NEW}] \\ \$\text{NULL} \end{array} \right\} \right] \quad [; \text{STACK} = \textit{stacksize}] \\ [; \text{DL} = \textit{dlsiz}] \end{aligned}$$

Parameters

<i>progfile</i>	Actual file designator of the program file that contains the prepared program.
<i>entrypoint</i>	Program entry point where execution is to begin. It contains a character string, terminated by a blank, specifying the entry point (label) in the program where execution is to begin when the program is executed. This point may be the primary entry point of the program, or any secondary entry point in the program's outer block. The default is the primary entry point.
NOPRIV	Declaration that program segments will be placed in nonprivileged (user) mode. This parameter is for programs prepared with privileged mode capability and makes them accessible to nonprivileged users. Normally, programs containing privileged instructions are executed in privileged mode only if the program was prepared with privileged mode (PM) capability. A program containing legally compiled privileged code, placed in nonprivileged mode, may abort when an attempt is made to execute it. If NOPRIV is specified in the RUN command, all program segments are placed in nonprivileged mode. Library segments are not affected because their mode is determined independently. NOPRIV produces the same effect as omitting the \$OPTION PRIVILEGED and OPTION PRIVILEGED entries in SPL source input. The default is that privileged mode programs remain in privileged mode.
LMAP	Request to produce a descriptive listing of the allocated (loaded) program on the file whose formal file designator is LOADLIST. If there is no reference to LOADLIST in any FILE command, the listing is sent to \$STDLIST. The default is no listing.
DEBUG	Request to issue a debug call before the first executable instruction of the program. Unless you have READ and EXECUTE access to the program, this parameter is ignored. If assigned privileged mode (PM) capability, you are put in privileged mode debug. If you do not have PM capability, you are put in user mode debug.

<i>segsz</i>	Maximum stack area (Z-DL) size permitted, in words. This parameter is included if it is expected that the size of DL-DB or Z-DB areas will be changed during program execution. Regardless of your specifications, MPE V/E may change <i>segsz</i> to accommodate table overflow conditions. If you prepare your program with <i>segsz</i> less than the configured minimum, the value is rounded up to the minimum or to the amount needed by the program (as calculated by the Segmenter). The maximum actual <i>segsz</i> permitted a program is 32,767 words. If your specified <i>segsz</i> exceeds the maximum, it is rounded down to 32,767 words.
<i>parameternum</i>	An integer containing a parameter to be passed to the new program, accessed through Q-4 of the outer block.
<i>stacksize</i>	Size of the initial local data area, Z-Q (initial), in the stack. This value, if specified, must be between 511 and 32,767 words. It overrides the default stack size estimated by the Segmenter.
<i>dlsz</i>	DL-DB area to be initially assigned to stack. This area is of interest mainly in programmatic applications. To accommodate system logging requirements, the DL-DB area is always rounded upward in such a way that the distance from the beginning of the stack data segment to the DB address is a multiple of 128 words. This value must be between -1 and 32,767 words. The default is estimated by the Segmenter.
G,P,or S	Searches the segmented procedure libraries of the program file's group and account. The G option first searches the group library, then the account library, then the system library. The P option searches the account library, then the system library. The S option searches the system library for external references to segmented procedures. The default is S.
NOCB	Request that file system not use the stack segment, PCBX, for its control blocks, even if sufficient space is available. This permits expansion of the stack, by using the DLSIZE and ZSIZE intrinsics, to the maximum possible limit at a later time. However, NOCB causes the file management system to operate more slowly.
<i>quotedstring</i>	A sequence of ASCII characters bounded by a pair of single quotation marks (apostrophes) or by double quotation marks. If you want a quotation to appear within <i>quotedstring</i>, the quotation and its quotation marks must also be bounded by quotation marks. For example, to insert "and" into a <i>quotedstring</i>, it must appear as ""and"". Similarly, 'and' must appear as "'and'". The maximum length of the string, including delimiters, is 255 characters. Refer to "Examples". The string is placed just after the global area of the new program's stack. A DB-relative byte pointer to the string in the new program's stack is placed at Q-5 of the stack, where Q is the initial value of the Q-register at activation time. The length of the string in bytes is placed at Q-6. If no string is specified, Q-5 and Q-6 both contain 0. For more information refer to the CREATEPROCESS intrinsic in the <i>MPE V Intrinsics Reference Manual</i> (32033-90007).

RUN

STDIN	Use of this parameter allows you to specify the file to be used as \$STDIN by the program being executed. If this parameter is omitted, or if nothing is specified after the “equals” sign, as in ;STDIN=" ", then STDIN defaults to the standard input device for the job or session. STDIN files should be kept unnumbered. This parameter may use one of the following subparameters: <table><tr><td><i>*formaldesig</i></td><td>The formal file designator for a file previously specified in a file equation.</td></tr><tr><td><i>fileref</i></td><td>The name of an existing permanent disk file.</td></tr><tr><td>\$NULL</td><td>The actual file designator of a system-defined file that is always treated as an empty file. When referenced by a program as \$STDIN, the program receives only an end-of-file indication when accessed. When referenced by a program as \$STDLIST, the associated write request is accepted by MPE V/E, but no physical output is actually performed. Thus, \$NULL can be used to discard unneeded output from an executing program.</td></tr></table>	<i>*formaldesig</i>	The formal file designator for a file previously specified in a file equation.	<i>fileref</i>	The name of an existing permanent disk file.	\$NULL	The actual file designator of a system-defined file that is always treated as an empty file. When referenced by a program as \$STDIN, the program receives only an end-of-file indication when accessed. When referenced by a program as \$STDLIST, the associated write request is accepted by MPE V/E, but no physical output is actually performed. Thus, \$NULL can be used to discard unneeded output from an executing program.
<i>*formaldesig</i>	The formal file designator for a file previously specified in a file equation.						
<i>fileref</i>	The name of an existing permanent disk file.						
\$NULL	The actual file designator of a system-defined file that is always treated as an empty file. When referenced by a program as \$STDIN, the program receives only an end-of-file indication when accessed. When referenced by a program as \$STDLIST, the associated write request is accepted by MPE V/E, but no physical output is actually performed. Thus, \$NULL can be used to discard unneeded output from an executing program.						
STDLIST	Use of this parameter allows you to specify the file to be used as \$STDLIST by the program being executed. If this parameter is omitted, or if nothing is specified after the equal sign, as in ;STDLIST=" ", then STDLIST defaults to the standard list device for the job or session. This parameter has the same subparameters as STDIN, but you may also specify the keyword NEW. <table><tr><td>NEW</td><td>The name to be assigned to a job or session by <i>fileref</i> is a temporary disk file created with the system defaults. The system defaults of the new file are fixed-length ASCII 132-byte records with a maximum file size of 1023 records.</td></tr></table>	NEW	The name to be assigned to a job or session by <i>fileref</i> is a temporary disk file created with the system defaults. The system defaults of the new file are fixed-length ASCII 132-byte records with a maximum file size of 1023 records.				
NEW	The name to be assigned to a job or session by <i>fileref</i> is a temporary disk file created with the system defaults. The system defaults of the new file are fixed-length ASCII 132-byte records with a maximum file size of 1023 records.						

Use

You may issue this command from a session or a job. You may not issue it in BREAK or from a program. Pressing **BREAK** suspends execution.

Operation

The RUN command executes a program prepared in a program file. It permits searching segmented libraries (SLs) to satisfy external references. Relocatable libraries (RLs) are not searched.

If the volume set containing the file to be run is not mounted, this command implicitly causes that volume set to be mounted.

If the file is temporary, only the logon group and account libraries for the current session are searched. Refer to *MPE File System Reference Manual* (30000-90236) for more information on file domains.

Examples

To list the segments of a loaded program, enter:

```
:RUN XLAB;LMAP
```

To run a program stored in the program file PROG4, beginning at the entry point SECLAB, enter:

```
:RUN PROG4,SECLAB
```

The following example runs a program TESTPROG with \$STDIN set to an old disk file named INPUT and \$STDLIST set to the line printer:

```
:FILE LPFILE;DEV=LP  
:RUN TESTPROG; MAXDATA=10000; STDIN=INPUT; STDLIST=*LPFILE
```

The next example runs a program using the STDIN parameter, setting \$STDIN to an existing disk file named INPUT, this time referenced through a file equation. In this example, \$STDLIST is set to a temporary disk file named RESULTS that is automatically created by the RUN command:

```
:FILE INFILE=INPUT,OLD  
:RUN TESTPROG;DEBUG;STDIN=*INFILE;STDLIST=RESULTS,NEW
```

The following example of the RUN command uses the INFO= parameter to pass a string to the program:

```
:RUN MYPROG;MAXDATA=2000;INFO= "A test with ""and"" characters"
```

In *quotedstring*, “and” is bounded by an extra pair of quotation marks. As a result, the string passed to the program is:

```
A test with "and" characters
```

SAVE

Saves a file in the permanent system file domain.

Syntax

$$\text{SAVE} \left\{ \begin{array}{l} \$\text{OLDPASS}, \text{newfilereference} \\ \text{tempfilereference} \end{array} \right\}$$

Parameters

\$OLDPASS	A system-defined temporary file. After this file is saved, it can no longer be referenced by the name \$OLDPASS.
<i>newfilereference</i>	New actual file designator assigned to \$OLDPASS when it is made permanent. Its format is: <i>filename[/lockword][.groupname[.acctname]]</i> If you use <i>groupname</i> , it must indicate a group to which you have SAVE access, as defined by your account manager. If you omit <i>groupname</i> , the logon group is assigned.
<i>tempfilereference</i>	Actual file designator of the temporary file to be made a permanent file under the same designator. The file is deleted from the job/session temporary file domain and entered in the system file domain. Its format is: <i>filename[/lockword][.groupname[.acctname]]</i> If you use <i>groupname</i> , it must indicate a group to which you have SAVE access, as defined by your account manager. If you omit <i>groupname</i> , the logon group is assigned.

Use

You may issue this command from a session, job, program, or in BREAK. Pressing **BREAK** has no effect on it.

Operation

The SAVE command saves a temporary file by converting it to a permanent file in the system file domain. This command is necessary when the subsystem or program that created your file does not allow you to save it while the program is executing. The volume set for the group in which the file is to be saved must be mounted, because files cannot be saved across volume sets.

You must specify a new file name for **\$OLDPASS** because MPE V/E does not allow **\$OLDPASS** as a permanent file name. If there is a file in the temporary domain that already has the name specified in *newfilereference*, MPE V/E attempts to save **\$OLDPASS** by creating a new temporary file. This temporary file name, created by **SAVE**, starts with **S** and is followed by seven digits, in the form:

Sdddhhmm

where *ddd* is the Julian day of the year, *hh* is the hour of the day, and *mm* indicates minutes. The new temporary file is then saved under the file name specified by *newfilereference*, and deleted from the temporary domain. If both temporary and permanent files exist under the name specified by *newfilereference*, the temporary **SAVE** file is saved as *Sdddhhmm*. In this case, a printed error message states the file name for the new **SAVE** file in the format *Sdddhhmm*. You can rename it later using **RENAME**.

You can use this command only for temporary files on disk. It is similar to opening a file with the **FOPEN** intrinsic, and then closing it with the **FCLOSE** intrinsic, using a permanent file disposition.

Examples

To save the temporary file **\$OLDPASS**, containing an object program, to the program file **PROGFILE**, enter:

:SAVE \$OLDPASS,PROGFILE

To save the temporary file **TEMPFL** as a permanent file with the same name, enter:

:SAVE TEMPFL

To save the temporary file **DATAFILE** in the group **GROUPX**, enter:

:SAVE DATAFILE.GROUPX

To save a temporary file (other than **\$OLDPASS**) and change its name, use the **SAVE** and **RENAME** commands. The system only searches the logon group and account directories in the current session. For example:

:SAVE DATAFILE

:RENAME DATAFILE,DATABASE

SECURE

Restores security provisions for a file that were previously suspended by a **RELEASE** command.

Syntax

SECURE *filereference*

Parameter

filereference Actual file designator of the disk file whose security provisions are to be restored. Its format is:

filename[/lockword] [.groupname[.acctnam]]

Use

You may issue this command from a session, job, program, or in **BREAK**. Pressing **BREAK** suspends execution.

Operation

The **SECURE** command restores all security provisions for a file that were previously suspended by a **RELEASE** command in this or another job.

You can only use this command for a permanent disk file whose file label identifies the user as the creator of that file. The **SECURE** command fails if the group's home volume set is not mounted. When the normal (default) MPE V/E security provisions are in effect, the file must belong to the logon account and group.

Example

To restore the security provisions previously in effect for the file named **TEXT1**, enter:

:SECURE TEXT1

SEGMENTER

Calls the MPE Segmenter.

Syntax

```
SEGMENTER [ listfile ]
```

Parameter

listfile Actual file designator of any ASCII output file that is to receive listed output from the Segmenter. The formal file designator is **SEGLIST**. The default is **\$STDLIST**. Usually this file is a line printer and must be defined in a **FILE** command, backreferenced as follows:

```
:FILE LISTFL;DEV=LP  
:SEGMENTER *LISTFL
```

Note



The formal file designator used in this command, **SEGLIST**, cannot be backreferenced as an actual file designator in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the **FILE** command.

Use

You may issue this command from a session or a job. You may not issue it in **BREAK** or from a program unless you or the Segmenter have process handling (PH) capability. Pressing **BREAK** suspends execution.

Example

To call the Segmenter from a session and transmit the output to a line printer instead of the standard list device, enter:

```
:FILE LISTFL;DEV=LP  
:SEGMENTER *LISTFL
```

Note



You must have **READ** and **LOCK** capability to use a relocatable library with the **SEGMENTER** command.

Additional Discussion

MPE Segmenter Reference Manual (30000-90011)

SET

Allows a job using a spooled **\$STDLIST** to mark its standard list device for deletion when the job terminates.

Syntax

$$\text{SET STDLIST} = \left\{ \begin{array}{l} \text{DELETE} \\ \text{SAVE} \end{array} \right\}$$

Parameters

DELETE	Flags the job's \$STDLIST for deletion at job termination.
SAVE	Cancels the effect of a previous SET STDLIST=DELETE command. The default is SAVE .

Use

You may issue this command from a job or a program. You may not issue it from a session or in **BREAK**. Pressing **(BREAK)** has no effect on it.

Operation

You can place the **SET** command anywhere between the **JOB** and **EOJ** statements. It is most practical to place it at the end of a job stream, since the command does not execute if the job fails. **\$STDLIST** then prints, allowing you to study your listing and locate the problem. You can reverse the effect of a **SET STDLIST=DELETE** by entering **SET STDLIST=SAVE** in the job stream. The **SET** command works only on jobs with a spooled **\$STDLIST**.

Example

```
!JOB EXAMPLE, MAC.TECHPUB,XGROUP
!CONTINUE
!RUN UPDATE.PUB.SYS;PARM=1;MAXDATA=16000
!IF JCW < FATAL THEN
!SET STDLIST=DELETE
!ENDIF
!EOJ
```

SETCATALOG

Causes the command interpreter to search a catalog of user- defined commands (UDCs) and establish an entry for each command, or to deactivate previously set catalogs.

Syntax

```
SETCATALOG [ catfilename [ /lockword ] [ , catfilename [ /lockword ] ] ... ] ... [ ;SHOW ]
[ ; { SYSTEM
    ACCOUNT } ] [ ;USER=user [ .acct ] ]
```

Parameters

catfilename Catalog file name. Usually, this is a file consisting of one or more user-defined commands (UDCs). When more than one user-defined command resides in the catalog file, the commands must be separated from each other by one line, the first character of which must be an asterisk. If you omit *catfilename*, all UDCs are deactivated at the level (user, account or system) specified in the SETCATALOG command.

Note



If you want to use two different UDC files, you must specify both files. If, for example, *udcfile1* is currently active at the user level and you want to add the UDCs in *udcfile2* as well, you must issue the command SETCATALOG *udcfile1,udcfile2*. If you specify *udcfile2* only, you deactivate *udcfile1*.

<i>lockword</i>	An optional lockword assigned by the creator of the file.
SHOW	Lists catalogs and UDCs as the user-defined commands are initialized. This parameter is useful for listing additional information if there is an error in UDC initialization. The SHOW command lists each UDC as it is checked. If an error occurs, it is listed after the faulty UDC.
ACCOUNT or SYSTEM	Specifies the level at which the catalog file is being defined or deactivated. ACCOUNT specifies that the catalog file is at the account level. It requires account manager (AM) capability. SYSTEM specifies that the catalog file is at the system level. It requires system manager (SM) capability. The default is the user level.
<i>user</i> [. <i>acct</i>]	The user name of the user whose catalog file is defined or reset. Requires account manager (AM) capability.

Use

You may issue this command from a session, job or in BREAK. You may not issue it from a program. Pressing **BREAK** has no effect on it.

SETCATALOG

Operation

The SETCATALOG command causes the command interpreter to search a specially prepared disk file for user-defined commands. The command interpreter then establishes a directory entry for each command in the file.

You must have READ and LOCK access to the UDC file in order to use the SETCATALOG command and to log on while that UDC file is in effect. Therefore, for account and system-level UDCs, the account and group file security under which the files reside must allow READ and LOCK access to all users. Lockwords that are specified using SETCATALOG are put into COMMAND.PUB.SYS, so you need not know the lockword to log on.

If you enter the SETCATALOG command with no parameters, all your user-level UDC files are removed from the system catalog. The files are not purged, but are deleted from the catalog. In order to purge or modify a particular UDC file, you must first remove it from the system UDC catalog by issuing the SETCATALOG command with a parameter list that does not include that particular UDC file name.

Example

To create a UDC file named MYCMNDS with the Editor, and have the command interpreter establish a directory entry for it, enter:

```
:EDITOR
HP32201A.7.15 EDIT/3000  WED, AUG 16, 1989,  8:01 AM
(C) HEWLETT-PACKARD CO. 1985

/ADD

    1 S                                ** UDC name          **
    2 SHOWJOB                        ** MPE V/E command   **
    3 **                               ** Separate UDCs **
    4 R                                ** UDC name          **
    5 RESUME                          ** MPE V/E command   **
    6 //

...
/KEEP MYCMNDS
/E
_

END OF SUBSYSTEM
:SETCATALOG MYCMNDS;SYSTEM  ** Establishes a UDC directory at the **
                               ** system level for both commands.**
                               ** You require SM capability.**
```

Now, **S** and **R** are established as system-defined commands. If you enter **S**, the **SHOWJOB** command is executed; if you enter **R**, the **RESUME** command is executed.

If, as account manager, you want to establish directory entries in the UDC file for users within your own account, enter:

:SETCATALOG NEWCMNDS;ACCOUNT **** Requires AM capability ****

Additional Discussion

Refer to chapter 3 of this manual.



SETDUMP

Enables the stack dump facility and initiates it at the time of an abort.

Syntax

$$\text{SETDUMP} \left[\left\{ \begin{array}{c} \text{DB} \\ \text{ST} \\ \text{QS} \end{array} \right\} [, \dots] \right] [; \text{ASCII}]$$

Parameters

- DB, ST, or QS** Dumps memory from the stack marker trace and registers at the time of abort in a specified job. **DB** dumps memory from DL to Q (initial) address. **ST** dumps memory from Q (initial) to S address. **QS** dumps memory from Q-63 to S address. The **QS** parameter is ignored if the **ST** parameter is used. The default is a display of the stack marker trace and all registers at the time of an abort.
- ASCII** Displays ASCII conversion of octal values requested by **DB**, **ST**, or **QS** parameters, along with octal display, but only in a job. The default is octal.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on it.

Operation

The **SETDUMP** command allows a display of the stack markers and all registers requested at the time of an abort. The stack marker trace is written to **\$STDLIST**. In an interactive session, all parameters are ignored, but if you have **READ** and **WRITE** access to the program file, **SETDUMP** automatically initiates **DEBUG**.

Examples

To use **SETDUMP** in an interactive session, enter:

:SETDUMP

Run the program **ABC.MAC.TECHPUBS**:

:RUN ABC.MAC.TECHPUBS

If the program ABC contains an error, it aborts, writes the stack marker trace to the current list device, enters DEBUG, and displays the following error message:

```

ABORT :ABC.MAC.TECHPUBS.%O.%763
PROGRAM ERROR #18 :PROCESS QUIT .PARAM = 2
*** ABORT STACK ANALYSIS ***

S=001154    DL=177650    Z=003425
Q=001160 P=000763  LCST= 000  STAT=U,1,1,L,0,1,CCG  X=000035
Q=001153 P=177777  LCST= S026  STAT=P,1,0,L,0,0,CCG  X=000000

*DEBUG* PRIV.0.764
?EQ          ** Terminates DEBUG **

PROGRAM TERMINATED IN AN ERROR STATE.  (CIERR 976)
:
```

To initialize the stack dump facility from a batch job and display the memory area from the Q (initial) to S address, with accompanying ASCII conversion of the octal data, add the following line to your stream job:

```
!SETDUMP ST;ASCII
```

Additional Discussion

MPE Debug/Stack Dump Reference Manual (30000-90012)

SETJCW

Creates or assigns a value to a job control word (JCW).

Syntax

$$\text{SETJCW } jcwname \text{ delimiter value } \left[\left\{ \begin{array}{c} + \\ - \end{array} \right\} value \right]$$

Parameters

<i>jcwname</i>	The name of a new or existing user or system-defined job control word (JCW). You can use “@” to specify all currently defined JCWs. You may not specify the following system-reserved JCWs: HPMINUTE, HPHOUR, HPDAY, HPDATE, HPMONTH, or HPYEAR.
<i>delimiter</i>	One or more punctuation characters or spaces, except “%” and “-”. Whatever character you use delimits the name and value.
<i>value</i>	One of the following: 1. An octal number between 0 and %177777, inclusive. 2. A decimal number between 0 and 65535, inclusive. 3. An MPE V/E-defined mnemonic for a JCW value (OK for 0; WARN for 16,384; FATAL for 32,768; SYSTEM for 49,152) or an offset value of a mnemonic (OK3, etc). 4. The name of an existing JCW. All specified values must be in the range of 0 to 65,535, inclusive. If you use the options “+” or “-”, the result of the indicated operation must also be within the range of 0 to 65,535, inclusive.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on it.

Operation

A job control word (JCW) is a flag that allows information to be passed between processes within a single job or session. There are three forms of JCWs: system-defined, user-defined, and system-reserved.

JCWs can be tested against specific values. You can program conditional statements that act according to the results of these tests. User-defined JCWs can also be set to user-selected values so that they reflect the completion of steps within the setting process. System-defined JCWs can be used to determine whether certain events have occurred within MPE V/E.

You can inspect the values in the system-reserved JCWs, but you cannot alter them.

The contents of a JCW may be displayed using the **SHOWJCW** command.

JCW Values and Mnemonics

JCWs may be assigned any positive integer value between 0 and 65,535 inclusive (%0 and %177777). MPE V/E treats these values as 16-bit unsigned integers, since it uses all 16 bits for numeric information, rather than using the most significant bit as a sign bit.

MPE V/E treats the two most significant bits of a JCW in a special way: the bits define “bases” or “steps” of 16K each. Each of these steps is given a mnemonic to simplify references to it or to the numbers between steps. If the 14 least significant bits are considered to be zeros, the two “step” bits, step value (in decimal), and mnemonic have the following relationship:

Bit Value	Step Value	Mnemonic
00	0	OK
01	16,384	WARN
10	32,768	FATAL
11	49,152	SYSTEM

It is important to remember that these mnemonics are not the names of JCWs. They cannot be used as user-defined JCW names.

You may use a combination of mnemonics and numbers to indicate numeric values between steps. If you specify a mnemonic and a number with no intervening spaces, an implied addition takes place. For example, **WARN3** has a value of 16,387, since it is **WARN** (16,384) plus 3. The value of the mnemonic plus the appended number value may not exceed 65,535. Again, no value of the form **mnemonic[number]** may be used as a valid user-defined *jcwname*.

You can also specify an explicit addition or subtraction using a “+” or “-” sign, as in **OK+7** (7) or **WARN-4** (16,380). You can also add a mnemonic to another mnemonic, as in **WARN+FATAL**. The result of a such mathematical operation must be in the range of 0 to 65,535 inclusive. If the number is out of range, an error message is generated and the value of the JCW remains unchanged. When the result of an operation is greater than the value of the next “step”, the JCW value displayed by the **SHOWJCW** command is the mnemonic of the higher step plus any offset. For example, the value **OK+16385** is displayed as **WARN1**.

User-defined JCWs

User-defined JCWs are created and initialized to a value by the **SETJCW** command or **PUTJCW** intrinsic. The JCW name contains alphanumeric characters and must begin with an alphabetic character. The name can be up to 255 characters long. The value assigned to the JCW must be in the range of 0 to 65,535, inclusive.

The **SETJCW** command scans MPE V/E’s JCW table for the name of the specified JCW (*jcwname*). If the specified name is found, the JCW is set to *value*. If the *jcwname* is not found, it is created and set to *value*. The term “*value*” as used here means the explicitly stated or the computed value.

You may not begin a JCW name with the mnemonic names **OK**, **WARN**, **FATAL**, or **SYSTEM** unless you append a number to the mnemonic such that the computed value exceeds 65,535; for example, **WARN999999**, or **SYSTEM20000**. If the computed value exceeds 65,535, MPE V/E

SETJCW

does not recognize the term as a valid mnemonic, and treats it as the name of a JCW. This restriction is intended to eliminate the possibility of an ambiguous JCW assignment. For example, it is unclear from the following two commands whether the JCW X is equal to 100 or to 0:

```
:SETJCW OK=100
:SETJCW X=OK
```

If you name a JCW with a mnemonic or predefined JCW value, you receive an error message, as in the following example:

```
:SETJCW OK200=1982
```

JCNAME CANNOT BE A VALID JCW VALUE (CIERR 1725)

Negative or out-of-range JCW values cause the following error message to be displayed:

VALUE NOT IN RANGE--LEGAL RANGE IS 0 TO 65535 (CIERR 1712)

System-defined JCWs

MPE V/E has two system-defined JCWs that are created for each job and session: JCW and CIERROR. The JCW named JCW is always initialized to zero at the beginning of the job or session and remains zero unless fatal errors occur, or unless you change the value. There are two special values for the system-defined JCW:

%140000 (System 0) Program aborted per user request.

>%140000 Program terminated in an error state.

The CIERROR JCW tracks command interpreter (CI) errors. CIERROR is set to zero at the beginning of the job or session. If a command interpreter error occurs, CIERROR is updated to reflect the current CI error message number. A valid command does not reset CIERROR to zero.

You are advised not to alter the values of the CIERROR and JCW job control words. Create user-defined JCWs for information you wish to control.

The following example shows the use of the CIERROR JCW:

```
:LISTTF
~
UNKNOWN COMMAND NAME. (CIERR 975)
:SHOWJCW CIERROR
CIERROR = 975
:RUN
~
NO PROGRAM FILE SPECIFIED. (CIERR 600)
:SHOWJCW CIERROR
CIERROR = 600
:LISTF ABC
ABC
:SHOWJCW CIERROR
CIERROR=600
:
```

System-Reserved JCWs

The system-reserved JCWs are HPMINUTE, HPHOUR, HPDAY, HPDATE, HPMONTH, and HPYEAR. They contain system-assigned minute, hour, day, date, month, and year information. If you attempt to assign values, an error message is displayed. You can retrieve the values in these JCWs with the **FINDJCW** intrinsic. The values can also be tested if the JCW is used with an **IF** command. The names of the system-reserved JCWs are reserved.

The following lists system-reserved JCWs and possible values:

HPDAY	Day of the week. The possible integers are 1 through 7. Sunday is indicated by 1. Saturday is indicated by 7.
HPDATE	Day of the month. The possible integers are 1 through 31.
HPMONTH	Month of the year. The possible integers are 1 through 12. January is indicated by 1.
HPYEAR	Year of the century. The possible integers are 00 through 99.
HPHOUR	Hour of the day. The possible integers are 0 through 23.
HPMINUTE	Minute of the hour. The possible integers are 0 through 59.

Conditional Execution Using JCWs

JCWs are usually used to control the flow of batch jobs, based on events that take place within the job. You can use **IF**, **THEN** and **ELSE** statements to test JCW values.

The following example illustrates a conditional execution. The sample job runs a program that edits, verifies, and counts valid transactions (**CHEKPROG**). If no fatal errors occur, the job runs the program **SHIPPROG**, which schedules shipments. The job then runs **FINALRPT**, which produces a final report. If fatal errors occur, **CHEKPROG** sets the value of the JCW **CHEKPROGSTAT** to **FATAL**, and **SHIPPROG** is not run. Instead, **ERRORRPT** is run, which produces an error report. A final report is also produced by **FINALRPT**.

You can display the contents of a system-defined JCW with the **SHOWJCW** command only if you specify the *jcwname*. Otherwise, **SHOWJCW** does not display system-defined JCWs.

```

!SETJCW CHEKPROGSTAT=OK
!CONTINUE                                ** Prevents abort in case of errors. **
!RUN CHEKPROG                            ** Edit,verify,and count valid transactions. **
! IF CHEKPROGSTAT<FATAL THEN             **If no fatal errors,schedule shipments.  **
!   RUN SHIPPROG                         ** Schedule shipments. **
! ELSE
!   SHOWJCW CHEKPROGSTAT
!   RUN ERRORRPT                         ** Produce error report and reset JCW to 0. **
! ENDIF
!RUN FINALRPT                           ** Produce final report.  **

```

SETJCW

Examples

To set the job control word CURR1 to 100, enter:

```
:SETJCW CURR1,100      ** The comma (,) is the delimiter **  
                        ** and is used as an "equals" sign. **
```

To set CURR1 to the value of the mnemonic WARN , enter:

```
:SETJCW CURR1/WARN    ** The slash (/) is the delimiter **  
                        ** and is used as an "equals" sign. **
```

To use an arithmetic operation to set a JCW, enter:

```
:SETJCW XX=5+6
```

To schedule a full backup job on Saturdays and a partial backup job on the other days of the week, enter:

```
:SETJCW FRIDAY=6  
:IF HPDAY = FRIDAY THEN  
    SCHEDJOB FULLBKUP;IN=1  
:ELSE  
    SCHEDJOB PARTBKUP;IN=1  
:ENDIF
```

SETMSG

Enables or disables the receipt of user or operator messages at the standard list device.

Syntax

$$\text{SETMSG } \left\{ \begin{array}{l} \text{OFF} \\ \text{ON} \end{array} \right\}$$

Parameters

- | | |
|-----|--|
| OFF | Sets job or session to QUIET mode and blocks the receipt of TELL command messages from other users. However, you can still send messages to the system console via TELLOP. |
| ON | Enables user or operator messages to be received and displayed at the standard list device. |

Use

You may issue this command from a session, job, in BREAK, or from a program. Pressing **BREAK** has no effect on it.

Operation

The SETMSG command allows a job or session to receive or block TELL messages from other users. WARN messages from the system operator override QUIET mode and are received and displayed. You can also send messages to the device designated as the system console, when it is running in QUIET mode, by using the TELLOP command. You can use the TELL command to send messages to sessions only; use the TELLOP command to send messages to devices.

Example

To block messages, enter:

:SETMSG OFF

To receive messages, enter:

:SETMSG ON

SHOWALLOCATE

Displays status information about **ALLOCATE** and/or **AUTOALLOCATE**.

Syntax

```
SHOWALLOCATE [ STATUS [ , listfile ]  
               [ ALLOCATE [ , [ fileset ] [ , listfile ] ]  
               [ AUTO [ , [ fileset ] [ , listfile ] ]  
               [ [ ALL ] [ , [ fileset ] [ , listfile ] ] ] ]
```

Parameters

STATUS	Request to display a summary of status information that includes: 1. Status of AUTOALLOCATE, either ON or OFF. 2. Number of programs allocated. 3. Number of programs autoallocated. 4. Size and percentage of utilization of the following system tables: Code segment table, code segment table extension, CSTX table and loader segment table.								
ALLOCATE	Request to display allocated programs specified by <i>fileset</i> and the number of users sharing each program.								
AUTO	Request to display autoallocated programs specified by <i>fileset</i>, and the number of users sharing each program.								
ALL	Request to display all information provided by the parameters STATUS, ALLOCATE and AUTO. This is the default parameter.								
<i>fileset</i>	Specifies the set of files to be searched for. The default is @.@.@. The format of this parameter is: <i>filedesignator</i>[.<i>groupdesignator</i>][.<i>acctdesignator</i>]] The fileset can be entered in any of the following formats and may use wildcard characters in any order as replacements. <table><tr><td>file.gp.acct</td><td>Showallocate file name in specified group and account.</td></tr><tr><td>file.group.®</td><td>Showallocate file name in specified group, any account.</td></tr><tr><td>file.®.®</td><td>Showallocate file name in any group and any account.</td></tr><tr><td>®.gp.acct</td><td>Showallocate all files in specified group and account.</td></tr></table>	file.gp.acct	Showallocate file name in specified group and account.	file.group.®	Showallocate file name in specified group, any account.	file.®.®	Showallocate file name in any group and any account.	®.gp.acct	Showallocate all files in specified group and account.
file.gp.acct	Showallocate file name in specified group and account.								
file.group.®	Showallocate file name in specified group, any account.								
file.®.®	Showallocate file name in any group and any account.								
®.gp.acct	Showallocate all files in specified group and account.								

- @.@.account Showallocate all files in all groups in specified account.
- @.@.@ Showallocate all files in system. This is the default.
- @ Showallocate all files in all groups in all accounts.
- @.group Showallocate all files in specified group, any account.

The characters “@”, “#”, and “?” can be used as wildcard characters in the *files* parameter. They have the following meanings:

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

These characters can be used as follows:

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

listfile Name of an output file to which all output is written. It is automatically specified as an ASCII file with variable-length records and the following characteristics: closed in a permanent domain, user-supplied carriage control (CCTL), OUT access mode, and EXC (exclusive access) option. This parameter may also be a backreferenced file. The default is \$STDLIST.

Use

You may issue this command from a session, job, program or in **BREAK**. Pressing **BREAK** suspends execution. You must have system supervisor (OP) capability to use this command.

Operation

The **SHOWALLOCATE** command generates the status information in the specified system tables and lists files that are allocated or autoallocated.

SHOWALLOCATE

Examples

To display status information for all allocated files in the system, enter:

```
SHOWALLOCATE ALLOCATE
```

ALLOCATED PROGRAMS	SHARE COUNT
-----	-----
EDITOR.PUB.SYS.....	0
FCOPY.PUB.SYS.....	2
LISTDIR5.PUB.SYS.....	1

```
NUMBER OF PROGRAMS FOUND = 3
```

To display status information for all autoallocated files starting with a character "S" in the SYS account, enter:

```
SHOWALLOCATE AUTO,S@.@.SYS
```

AUTOALLOCATED PROGRAMS	SHARE COUNT
-----	-----
SPOOK5.PUB.SYS.....	1
SLPATCH.PUB.SYS.....	0

```
NUMBER OF PROGRAMS FOUND = 2
```

To display summary status information regarding allocation, enter:

```
SHOWALLOCATE STATUS
```

```
ALLOCATION STATUS:
```

```
AUTOALLOCATE IS ON
NUMBER OF PROGRAMS ALLOCATED    =  3
NUMBER OF PROGRAMS AUTOALLOCATED =  7
```

ALLOCATION RELATED TABLES	SIZE	% USED
-----	----	-----
CODE SEGMENT TABLE	191	52
CODE SEGMENT TABLE EXTENSION	1023	9
CSTX BLOCK TABLE	144	13
LOADER SEGMENT TABLE	32000	3

SHOWALLOW

Displays the operator commands that have been allowed.

Syntax

```
SHOWALLOW { user.acct
            user.@
            @.acct
            @.@ }
```

Parameters

user Defines a particular user.

acct Defines a particular account.

@ All users, if used in place of *user*, or all accounts, if substituted for *acct*.
The default is that the logged-on user, account, and group are displayed.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** aborts execution. You must have account manager (AM) or system manager (SM) capability to execute this command for other groups or accounts.

Operation

The **SHOWALLOW** command displays the operator commands that are allowed to specific users if the *user.acct* form is entered. If you enter the @.@ form, the commands allowed to all users in all accounts are displayed. You require system manager (SM) capability to use the @.@ form; account managers (AM capability) may specify all users in their own account. When **SHOWALLOW** is executed from the system console, you may substitute “@” for *user* and/or *acct*. In addition, **SHOWALLOW** separately lists which operator commands have been globally allowed.

Example

To list the operator commands allowed to the user **MAC,PUB.SYS**, enter:

```
:SHOWALLOW USER.SYS
#S86 USER.SYS
      USER HAS THE FOLLOWING COMMANDS ALLOWED:
ABORTIO          ACCEPT          DOWN          GIVE

      THERE ARE NO GLOBAL ALLOWS DEFINED.
```

Additional Discussion

Refer to the **ALLOW** command.

SHOWCACHE

Displays a summary of disk caching performance. Disk caching software is not included in the fundamental operating system, and must be purchased separately.

Syntax

SHOWCACHE

Parameters

None.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on it.

Operation

The **SHOWCACHE** command summarizes the performance of disk caching on your system, displaying nine columns of information (shown below). Since it only allows you to look at disk caching performance, and prohibits the modification of caching parameters, any user may execute **SHOWCACHE**.

:SHOWCACHE

DISC	CACHE	READ	WRITE		PROCESS		% OF CACHE	
LDEV	REQUESTS	HIT%	HIT%	READ%	STOPS	K-BYTES	MEMORY	DOMAINS
1	21529	93	85	82	1705	686	11	218
2	35093	77	59	69	6155	2760	45	675
Total	56622	83	65	74	7860	3446	57	893

62% of user I/Os eliminated.
Data overhead is 229K bytes.
Sequential fetch quantum is 96 sectors.
Random fetch quantum is 16 sectors.
Block on Write = NO.

Each column of information provides the following information:

DISC LDEV	Logical device number(s) of the disk(s) being cached. Read across the row to see the other entries on a particular disk. The bottom row is a total across all disks.
CACHE REQUESTS	Tally of the number of logical disk I/O requests against caching. This is a double-word counter and resets to zero automatically when caching is started on the disk, or when the counter overflows.

READ HIT%	Cumulative percentage of logical disk I/O on read requests: read requests that were satisfied by data found in a main memory cache domain when the read request occurred.
WRITE HIT%	Cumulative percentage of logical disk I/O on writes requests: write requests that were accomplished in a main memory cache domain when the write request occurred.
READ%	Percentage of logical disk I/Os in the system which were read requests.
PROCESS STOPS	Number of times a process stopped in order to complete a cache I/O request. Process Stops occur due to read misses, or due to write hits in which the cached region is currently being posted to disk. This is a double-word counter which resets to zero if it overflows.
K-BYTES	Count of the number of thousands of bytes (1024-byte groups) which are currently being used for main memory cache domains.
% OF MEMORY	Percentage of the total memory in the machine which is currently allocated for use by cache domains.
CACHE DOMAINS	Number of cache domains in memory.

In addition, the **SHOWCACHE** command displays information about data overhead, which is the current dynamic data structure overhead used to support disk caching. The “sequential fetch” and “random fetch” quantums, established by default or selected with the **CACHECONTROL** command, specify the number of sectors read from the disk for sequential and random access files respectively. **BLOCKONWRITE**, which is also set with the **CACHECONTROL** command, determines whether or not MPE V/E blocks the process until posting of the cache buffers to disk is complete.

Additional Discussion

The **SHOWCACHE** command does not require any special capabilities. However, you are allowed only to look at disk caching performance and not to change any of the options or parameters. If you want to change the disk caching performance, refer to the **STARTCACHE**, **STOPCACHE**, and **CACHECONTROL** commands in this manual. Refer also to the **FSETMODE** intrinsic in *MPE V Intrinsic Manual* (32033-90007).



SHOWCATALOG

Lists user-defined command (UDC) files.

Syntax

```
SHOWCATALOG [ listfile ;USER=user[.acct] ]
```

Parameters

- listfile* An arbitrary file name, used to identify the UDC file when the output of SHOWCATALOG is sent to a printer. The output may be redirected with a previous **FILE** command. The default is that UDC files are displayed on your terminal (the \$STDLIST device).
- user*[*.acct*] A particular user whose UDC file names are displayed on the screen or sent to a list file; the contents of the files are not displayed.
- The wildcard characters “@”, “#”, and “?” may be used in *user*[*.acct*] as you would use them in the specification of any user or any account, where:
- | | |
|---|--|
| @ | One or more alphanumeric characters. Used by itself, “@” denotes all members of the set. |
| # | One numeric character. |
| ? | One alphanumeric character. |
- Each wildcard character counts toward the eight-character limit for *user* or *acct*.

Use

You may issue this command from a session, job, or in **BREAK**. You may not issue it from a program. Pressing **BREAK** aborts execution.

Operation

The SHOWCATALOG command lists user-defined command (UDC) files. Unless you redirect the output to another device with the **FILE** command, as shown in the example, the listing appears on your terminal. It indicates at which level (user, account, or system) the files have been defined, so that you may display the current catalog before adding or deleting system-level UDCs (if you are the system manager) or account-level UDCs (if you are account manager).

Examples

To list system, account, and user level UDC files on your terminal, enter:

```
:SHOWCATALOG
```

To store the current UDC file on disk as LFILE (which may also be printed) enter:

```
:BUILD LFILE
```

```
:FILE LFILE,OLD;DEV=DISC
```

```
:SHOWCATALOG *LFILE
```

Additional Discussion

Refer to the discussion of UDCs in chapter 3 of this manual.

SHOWCOM

Displays status information about a communication device.

Syntax

```
SHOWCOM ldev [ ;ERRORS ] [ ;RESET ]
```

Parameters

<i>ldev</i>	The logical device number of a communication system device.
ERRORS	Displays the full status list. If not specified, an abbreviated list is displayed.
RESET	Reset all status information to zero after it has been displayed.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on it. You may issue it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

Operation

The status information generated by the **SHOWCOM** command can be used to determine the activity and quality of a communication line. It includes:

- Number of messages sent and received.
- Last recoverable and irrecoverable errors.
- Number of recoverable and irrecoverable errors.
- Number of retransmissions, response timeouts, clear-to-send losses, and underruns.
- Number of check character (BCC/CRC) errors, receive timeouts, carrier losses, and overruns.
- Line state (closed, connected or disconnected).

Note that clear-to-send losses and carrier losses are valid only for communication lines configured as full duplex.

When the line is open, connected or disconnected, the status information displayed is a duplicate of the statistics generated from the last opening of the line until the **SHOWCOM** command is invoked. When the line is closed, the status information obtained at the close of the line reflects the statistics generated during the last open/close sequence.

Examples

To display the status of logical device #18, enter:

:SHOWCOM 18

```

                LDN - 18
MESSAGES SENT      0          MESSAGES RECVD 0
      LAST RECOVERABLE ERROR      0
      LAST IRRECOVERABLE ERROR    201
      LINE IS CLOSED

```

To display the full status of logical device #18, enter:

:SHOWCOM 18;ERRORS

```

      TRANSMIT                LDN - 18                RECEIVE
MESSAGES SENT      0          MESSAGES RECVD 0
RETRANSMISSIONS    0          BCC/CRC ERRORS 0
RESPONSE TIMEOUTS  0          RECV TIMEOUTS 0
UNDERRUNS          0          OVERRUNS      0
CLR TO SEND LOSSES 0          CARRIER LOSSES 0
      # OF RECOVERABLE ERRORS
      LAST RECOVERABLE ERROR
      # OF IRRECOVERABLE ERRORS
      LAST IRRECOVERABLE ERROR
      LINE IS CLOSED

```

Note



The "LAST IRRECOVERABLE" and "LAST RECOVERABLE" error codes are specified in the *Communications Handbook* (30000-90105).

SHOWDEV

Reports the status of input/output devices and information about any associated access control definitions (ACDs).

Syntax

```
SHOWDEV  $\left[ \begin{array}{l} ldev \\ classname \end{array} \right] [ ; ACD ]$ 
```

Parameters

- ldev* Logical device number of device for which status information is to be displayed. This number is unique for each device. The default is that status information for all system devices on the system is displayed.
- classname* - Device class name of device(s) for which status information is to be displayed. This name may apply to several devices. The default is that status information for all devices on the system is displayed.
- ACD Indicates that ACD information related to the requested device set is to be displayed. ACD information is only displayed if it exists for a device and if you are allowed access to the information.

Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** aborts execution.

Operation

The **SHOWDEV** command lets you display the status information for all input/output devices on the system. The display appears in the following format:

```
:SHOWDEV
LDEV AVAIL      OWNERSHIP      VALID      DEN      ASSOCIATION

  1  DISC      43 FILES
  6  SPOOLED  SPOOLER OUT
  8  AVAIL
20 A UNAVAIL  #S311:  2 FILES
```

The following items may appear in the listing, always displayed on the standard list device:

Column	Meaning
LDEV	Includes the logical device number and may include one of the following:
J	Accepts jobs.
D	Accepts data.
A	Accepts jobs and data.

AVAIL	Lists the availability of devices and disks as follows:
AVAIL	The device is available as a real, nonsharable device.
SPOOLED	The device is available via input or output spooling.
UNAVAIL	The device is not available (it is under the control of a job, session or a system process such as a spooler).
DISC	The device is a disk and is always available.
DISC (RPS)	The device is a CS-80 disk on which rotational position sensing (RPS) has been enabled.
OWNERSHIP	Includes device ownership and may include one of the following:
SYS	Controlled by the system. If # <i>nnn</i> appears, it specifies the process identification number (PIN) of the controlling process, or program.
SPOOLER IN	Input spooling in effect, controlled by spooler.
SPOOLER OUT	Output spooling in effect, controlled by spooler.
# <i>Jnnn</i>	Controlled by the indicated job.
# <i>Snnn</i>	Controlled by indicated session.
DIAG	The device has been allocated to diagnostic testing by the system operator via the GIVE command.
<i>nn</i> FILES	Indicates the number of files currently in use on a disk.
DOWN	The device is offline, requested by the system operator via the DOWN command.
DP	The device is being taken offline (there is a DOWN command operation pending).
VOLID	The volume identification; it may include one of the following:
IBM	The named magnetic tape volume that has a label written in the IBM format.
ANSI	The named magnetic tape volume that has a label written in the ANSI format.
NOLABEL	The named magnetic tape volume that has no label. This is the default.
DEN	Density of the tape; it may include one of the following:
6250	Density of 6250 BPI (bytes-per-inch).
1600	Density of 1600 BPI, or, the density of the tape is unrecognizable.
ASSOCIATION	Indicates the logical devices by device class which have been established by the user with the ASSOCIATE command.

SHOWDEV

Examples

To display the status of the device identified by logical device #5, enter:

```
:SHOWDEV 5
LDEV  AVAIL      OWNERSHIP      VOLID    DEN    ASSOCIATION

  5   SPOOLED    SPOOLER OUT
```

To display the status of all devices of the device class CARD, enter:

```
:SHOWDEV CARD
LDEV  AVAIL      OWNERSHIP      VOLID    DEN    ASSOCIATION

  6 A AVAIL
```

To display access control definitions (ACDs) associated with a device, enter:

```
:SHOWDEV 17;ACD

LDEV  AVAIL      OWNERSHIP      VOLID    DEN    ASSOCIATION
 17   SPOOLED    SPOOLER OUT
      ACD ENTRIES: JOE.DOE        : R
                      Q.Q         : X
```

SHOWIN

Reports the status of input device files.

Syntax

```
SHOWIN [ #Innn
        STATUS
        SP
        item[ ; item[ ; ... ] ] ]
```

Parameters

- #Innn** Identifier of the particular input device file for which information is to be displayed. The default is that MPE V/E displays information for all input device files used by the logon job or session.
- STATUS** Summarizes the status information for all current input device files. The default is that MPE V/E displays information for all input device files used by the logon job or session. The information appears in the following format:
- ```

 8 FILES
 0 ACTIVE
 0 READY; INCL 0 SPOOFLES, 0 DEFERRED
 8 OPENED; INCL 0 SPOOFLES
 0 SPOOFLES; 0 SECTORS
 0 LOCKED; INCLUDING 0 SPOOFLES
```
- SP** Displays the status information for the currently spooled input device files associated with the logon job or session. The default is a display of status information for all input device files.
- item** Displays the status of current input device files as identified. The default is that MPE V/E displays status information for all input device files used by this job. The syntax appears below.

## SHOWIN

### Syntax for *item*

$$\left[ \text{DEV} = ldev \right] \left[ \text{JOB} = \left\{ \begin{array}{l} @J \\ @S \\ @ \\ \left[ \# \right] Jnnn \\ \left[ \# \right] Snnn \end{array} \right\} \right] \left[ \left\{ \begin{array}{l} \text{ACTIVE} \\ \text{OPENED} \\ \text{READY} \end{array} \right\} \right]$$

### Parameters for *item*

*ldev* Displays the status of input device files identified by logical device number *ldev*.

*JOB=* Displays the status of input device files. *JOB=* may be one of the following options:

*@J* Displays the status of input device files for all jobs.

*@S* Displays the status of input device files for all sessions.

*@* Displays the device files for all jobs and sessions.

*[#] Jnnn* Displays the status of all input device files for a specified job.

*[#] Snnn* Displays the status of all input device files for a specified session.

The default is *@*.

*ACTIVE*,  
*OPENED*, or  
*READY* Displays the status of all input files in a specified state. *ACTIVE* displays the status of active device files. *OPENED* displays the status of opened device files. *READY* displays the status of ready device files.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** aborts execution.

### Operation

The **SHOWIN** command lets you display the status information about one or more currently defined input device files. This information reflects the status at the time you entered the command, and always appears on the standard list device. Except for the keyword **STATUS**, which has its own format (refer to “Parameters”), the format of the information is:

| DEV/CL | DFID   | JOBNUM | FNAME   | STATE  | FRM | SPACE | RANK | PRI | #C |
|--------|--------|--------|---------|--------|-----|-------|------|-----|----|
| 10     | #I1088 | #J133  | \$STDIN | OPENED |     |       |      |     |    |

The information displayed in this format is defined as follows:

| Column | Meaning                                                                                                                                                                                                                                                                                                |        |                                                   |       |                                                                   |        |                                        |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|---------------------------------------------------|-------|-------------------------------------------------------------------|--------|----------------------------------------|
| DEV/CL | Logical device number of device.                                                                                                                                                                                                                                                                       |        |                                                   |       |                                                                   |        |                                        |
| DFID   | Device file identification in the form #Innn.                                                                                                                                                                                                                                                          |        |                                                   |       |                                                                   |        |                                        |
| JOBNUM | Job or session number ( <i>jsnum</i> ) of the job or session using the device file, if not used for ready or active data. Otherwise, the job or session name appears on the line following the standard device information.                                                                            |        |                                                   |       |                                                                   |        |                                        |
| FNAME  | File name associated with the device file.                                                                                                                                                                                                                                                             |        |                                                   |       |                                                                   |        |                                        |
| STATE  | One of the following: <table> <tr> <td>ACTIVE</td><td>Input being read from a spooled device to a disk.</td></tr> <tr> <td>READY</td><td>Input spooling completed: file is now ready for use by a program.</td></tr> <tr> <td>OPENED</td><td>A file is being accessed by a program.</td></tr> </table> | ACTIVE | Input being read from a spooled device to a disk. | READY | Input spooling completed: file is now ready for use by a program. | OPENED | A file is being accessed by a program. |
| ACTIVE | Input being read from a spooled device to a disk.                                                                                                                                                                                                                                                      |        |                                                   |       |                                                                   |        |                                        |
| READY  | Input spooling completed: file is now ready for use by a program.                                                                                                                                                                                                                                      |        |                                                   |       |                                                                   |        |                                        |
| OPENED | A file is being accessed by a program.                                                                                                                                                                                                                                                                 |        |                                                   |       |                                                                   |        |                                        |
| FRM    | Not applicable to input files.                                                                                                                                                                                                                                                                         |        |                                                   |       |                                                                   |        |                                        |
| SPACE  | Approximate disk space currently used (in sectors), for input spoolfiles only.                                                                                                                                                                                                                         |        |                                                   |       |                                                                   |        |                                        |
| RANK   | Not applicable to input files.                                                                                                                                                                                                                                                                         |        |                                                   |       |                                                                   |        |                                        |
| PRI    | Not applicable to input files.                                                                                                                                                                                                                                                                         |        |                                                   |       |                                                                   |        |                                        |
| #C     | Not applicable to input files.                                                                                                                                                                                                                                                                         |        |                                                   |       |                                                                   |        |                                        |

## Examples

The following is an example of how to determine the status of an individual input device file:

:SHOWIN #I80

|        |      |        |         |        |     |       |      |     |    |
|--------|------|--------|---------|--------|-----|-------|------|-----|----|
| DEV/CL | DFID | JOBNUM | FNAME   | STATE  | FRM | SPACE | RANK | PRI | #C |
| 43     | #I80 | #S37   | \$STDIN | OPENED |     |       | 8    |     |    |

If you do not know the device file identification number (DFID) of the device file whose status you want to determine, you may request the status display by entering either the logical device number or the device class name of the device on which the file originated:

:SHOWIN DEV=43

|        |      |        |         |        |     |       |      |     |    |
|--------|------|--------|---------|--------|-----|-------|------|-----|----|
| DEV/CL | DFID | JOBNUM | FNAME   | STATE  | FRM | SPACE | RANK | PRI | #C |
| 43     | #I80 | #S37   | \$STDIN | OPENED |     |       |      |     |    |

## SHOWIN

You may also request displays of device file information using various combinations of qualifications (devices, jobs or sessions, and states). For example, to display information about all OPENED input device files used by all sessions (but not jobs) in the system, enter:

:SHOWIN JOB=QS;OPENED

| DEV/CL | DFID | JOBNUM | FNAME   | STATE  | FRM | SPACE | RANK | PRI | #C |
|--------|------|--------|---------|--------|-----|-------|------|-----|----|
| 7      | #I81 | #S38   | MASTER  | OPENED |     |       |      |     |    |
| 26     | #I36 | #S18   | \$STDIN | OPENED |     |       |      |     |    |
| 32     | #I85 | #S41   | \$STDIN | OPENED |     |       |      |     |    |
| 34     | #I58 | #S26   | \$STDIN | OPENED |     |       |      |     |    |
| 42     | #I64 | #S28   | \$STDIN | OPENED |     |       |      |     |    |
| 43     | #I80 | #S37   | \$STDIN | OPENED |     |       |      |     |    |
| 50     | #I84 | #S40   | \$STDIN | OPENED |     |       |      |     |    |
| 51     | #I35 | #S17   | \$STDIN | OPENED |     |       |      |     |    |

8 FILES (DISPLAYED):

0 SPOOFLES: 0 SECTORS

## SHOWJCW

Displays the current state of one or more job control words (JCWs).

### Syntax

SHOWJCW [*jcwname*]



### Parameters

*jcwname*      The name of a valid job control word (JCW). The default is that all user and system-defined JWCs are displayed.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** aborts execution.

### Operation

The SHOWJCW command is used to display the current state of one or more JCWs. If you specify a particular JCW (user or system-defined, or system-reserved), MPE V/E displays the state of that particular JCW. If you do not specify a particular JCW, user-defined and system-defined JCWs are displayed. The value of system-reserved JCWs is displayed only if you specifically enter the *jcwname*. You may retrieve the value assigned a JCW with the FINDJCW intrinsic, then test it with an IF command. In this way, the value of a given JCW can be used to conditionally execute another instruction or set of instructions. For example:

```
!CONTINUE
!SPL MYPROG,MYUSL
!IF JCW>=FATAL THEN
! TELL MAC.TECHPUBS;COMPILE FAILED
!ELSE
! TELL MAC.TECHPUBS;COMPILE COMPLETED
!ENDIF
```

### Examples

To show the current state of all user and system-defined JCWs, enter:

```
:SHOWJCW
JCW = 0
CIERROR = 0
```

To display the current state of a valid user-defined JCW named JCW1, enter:

```
:SHOWJCW JCW1
JCW1=3
```



## **SHOWJCW**

To display the contents of a system-reserved JCW, enter:

```
:SHOWJCW HPDAY
HPDAY=4
```

## **Additional Discussion**

Refer to SETJCW in this chapter.

## SHOWJOB

Displays the status information about jobs and/or sessions.

### Syntax

```
SHOWJOB [[#]Snnn
 [#]Jnnn
 STATUS
 SCHED
 item[;item[;...]]] [*listfile]
```

### Parameters

|           |                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [#]Snnn   | The number (assigned by MPE V/E) of the session for which the status information is to be displayed. The information appears in Type I format, described under "Operation". The default is that the status information for all jobs and sessions is displayed.                                                                                                                                                    |
| [#]Jnnn   | The number (assigned by MPE V/E) of the job for which status information is to be displayed. The information is in Type I format, described under "Operation". The default is that the status information for all jobs and sessions is displayed.                                                                                                                                                                 |
| STATUS    | Lists the number of jobs and sessions in each processing state and the current jobfence and job/session limits. This information is in Type II format, described under "Operation". The default is that the status information for all job and sessions is displayed.                                                                                                                                             |
| SCHED     | Displays only the scheduled jobs. The information is in Type III format, described under "Operation".                                                                                                                                                                                                                                                                                                             |
| *listfile | Formal file designator of the file on which the output listing is written. A backreference to a file equation is required. The <i>listfile</i> is a temporary file with a record size of 256 bytes, blocked one record per block, with carriage control (CCTL), and the time and date displayed. You can override the default characteristics of <i>listfile</i> with the FILE command. The default is \$STDLIST. |
| item      | A list of jobs and sessions whose status is displayed. The default is that the status information for all jobs and sessions is displayed. The syntax appears below.                                                                                                                                                                                                                                               |

## SHOWJOB

### Syntax for *item*

$$\left[ \text{JOB} = \left\{ \begin{array}{l} \text{QJ} \\ \text{QS} \\ \text{Q} \\ [\text{jsname}, ] \text{username.acctname} \end{array} \right\} \right] \left[ ; \left\{ \begin{array}{l} \text{INTRO} \\ \text{EXEC} \\ \text{SUSP} \\ \text{WAIT} \left[ \begin{array}{l} , \text{N} \\ , \text{D} \end{array} \right] \end{array} \right\} \right]$$

### Parameters for *item*

**JOB=** A list of jobs/sessions for which status information is to be displayed. Use one of the following options:

|                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>QJ</b>                          | Displays status information for all jobs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>QS</b>                          | Displays status information for all sessions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Q</b>                           | Displays status information for all jobs and sessions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>[jsname,].username.acctname</i> | The <i>jsname</i> is an optional name that you give to the session or job. The <i>username</i> is the user name established by the account manager. This name may consist of one to eight alphanumeric characters, beginning with an alphabetic character. The <i>acctname</i> is the name of the account established by the system manager. This name may consist of one to eight alphanumeric characters, beginning with an alphabetic character. You can use “@” to replace the <i>jsname</i> or <i>username</i> in a specified account. |

**INTRO, EXEC, SUSP, or WAIT** Displays the status of all jobs or sessions in a specified state. **INTRO** means that the job is introduced. In this case the spooler process validates the **JOB** command and, if the job is legitimate, copies the job input records to disk. **EXEC** means that the job is executing. **SUSP** means that the job or session is suspended, because table entries or system resources are unavailable. **WAIT** means that there are no available list devices for the job. **WAIT** has the following subparameters:

|          |                                                               |
|----------|---------------------------------------------------------------|
| <b>N</b> | Displays the status of nondeferred <b>READY</b> device files. |
| <b>D</b> | Displays the status of deferred <b>READY</b> device files.    |

If information for only one device file is displayed, output is in Type I format; if information for more than one device file is displayed, output is in Type I followed by Type II format. Format types are described under “Operation”.

## Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** aborts execution.

## Operation

The **SHOWJOB** command enables you to determine the number of jobs and sessions in each processing state, the current job fence and job/session limits, and allows you to keep track of individual spooled and streamed jobs that are entered in the system. The output appears in two possible formats or in a combination of both formats:

Type I:

```

JOBNUM STATE IPRI JIN JLIST INTRODUCED JOB NAME

#S16 EXEC 45 45 MON 7:08A TEST.PUBS

JOBFENCE= 0; JLIMIT = 3; SLIMIT= 16

```

Type II:

```

7 JOBS:
 0 INTRO
 0 WAIT; INCL 0 DEFERRED
 7 EXEC; INCL 7 SESSIONS
 0 SUSP
JOBFENCE= 0; JLIMIT= 3; SLIMIT= 16

```

If you use the **SHOWJOB SCHED** command, the output is displayed as shown below. The **STATE** field shows that the job is scheduled. The **SCHEDULED-INTRO** field shows the time and date at which the job is to be introduced to the system. The scheduled jobs are listed in the order in which they are to be introduced to the system. If you enter only the **SHOWJOB** command, the formatted output for jobs and sessions in the **INTRO**, **WAIT**, and **EXEC** states is displayed first in the Type I and Type II formats. The formatted data for jobs in the **SCHED** state is displayed last and is in the Type III format.

Type III:

CURRENT: 5/13/85 1600

```

JOBNUM STATE IPRI JIN JLIST SCHEDULED-INTRO JOB NAME

#J38 SCHED 3 10 6 5/16/84 11:24 NOTHING,JON.OSE
#J23 SCHED 8 10 PP 5/25/84 8:01 REPORT,MGR.OSE
#J25 SCHED 8 10 LP 7/4/84 18:05 FIREWORK,UNCLE.SAM

3 SCHEDULED JOB(S)

```

## SHOWJOB

### Examples

To determine the number of jobs and sessions in each processing state, the current job fence and the job and session limits, enter:

```
:SHOWJOB STATUS
```

```
6 JOBS:
```

```
0 INTRO
```

```
0 WAIT; INCL 0 DEFERRED
```

```
6 EXEC; INCL 6 SESSIONS
```

```
0 SUSP
```

```
JOBFENCE= 0; JLIMIT= 3; SLIMIT= 16
```

To get a report on all jobs and sessions in the system, enter:

```
:SHOWJOB
```

| JOBNUM | STATE | IPRI | JIN | JLIST | INTRODUCED | JOB NAME     |
|--------|-------|------|-----|-------|------------|--------------|
| #S745  | EXEC  |      | 29  | 29    | MON 2:53P  | DL,SPL.ALANG |
| #S746  | EXEC  |      | 26  | 26    | MON 2:53P  | CLI.AOPSYS   |

```
2 JOBS:
```

```
1 INTRO
```

```
0 WAIT; INCL 0 DEFERRED
```

```
2 EXEC; INCL 2 SESSIONS
```

```
0 SUSP
```

```
JOBFENCE= 2; JLIMIT= 1; SLIMIT= 16
```

## SHOWJOB

The following example of a SHOWJOB command sequence illustrates how you can override the default characteristics of *listfile* with the FILE command, and shows the output produced with the new *listfile* characteristics:

```
:FILE A;REC=40,1,F,ASCII;NOCCTL
:SHOWJOB;*A
:SAVE A
:FCOPY FROM=A;TO=
```

HP32212A.3.17 FILE COPIER (C) HEWLETT-PACKARD CO. 1982

MON, AUG 14, 1989, 7:54 AM

| JOBNUM | STATE | IPRI | JIN | JLIST | INTRODUCED | JOB NAME             |
|--------|-------|------|-----|-------|------------|----------------------|
| #S46   | EXEC  |      | 20  | 20    | MON 7:14A  | OPERATOR.SYS         |
| #S45   | EXEC  |      | 47  | 47    | MON 6:37A  | MAC.PUBS             |
| #S47   | EXEC  |      | 10S | LP    | MON 7:26A  | SUPPORT.DOC          |
| #S48   | EXEC  |      | 102 | 102   | MON 7:28A  | MAC.TECH             |
| #J19   | EXEC  |      | 28  | 28    | MON 6:41A  | JON.OSE              |
| #S49   | EXEC* |      | 34  | 34    | MON 7:31A  | FLASH.G              |
| #J21   | EXEC  |      | 10S | LP    | MON 7:15A  | DELIVER,MAIL.MAIL    |
| #J22   | EXEC  |      | 10S | LP    | MON 7:14A  | RSPPOOLJ,RSPPOOL.SYS |

8 JOBS (DISPLAYED):

0 INTRO

0 WAIT; INCL 0 DEFERRED

8 EXEC; INCL 5 SESSIONS

0 SUSP

JOBFENCE= 6; JLIMIT= 4; SLIMIT= 50

EOF FOUND IN FROMFILE AFTER RECORD 17

18 RECORDS PROCESSED \*\*\* 0 ERRORS

END OF SUBSYSTEM

:

The SHOWJOB command reports a job or session as being in EXEC\* when it is initializing. After initialization is complete, the state changes to EXEC. The number shown in the EXEC state is the sum of the jobs and sessions in both EXEC and EXEC\*.

---

## SHOWLOG

Displays the number of the system's current log file and the percentage of disk space used.

### Syntax

```
SHOWLOG
```

### Parameters

None.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on it. You must have system supervisor (OP) capability, or issue the command from the console unless it is distributed to users with the **ALLOW** command.

### Operation

The **SHOWLOG** command lets you display information about the system log file. The log file number, “*xxxx*”, and percentage of file space used, “*yy*”, are displayed in the format:

```
LOG FILE LOG xxxx IS yy% FULL
```

If the logging system is disabled, MPE V/E displays a “**NO LOGGING**” message. If logging is enabled but currently suspended due to an error, both messages appear.

### Example

To display the current log file status, enter:

```
:SHOWLOG
```

```
LOG FILE LOG2075 IS 20% FULL
```

### Additional Discussion

Refer to the discussion of system logging in “MPE Logging Facilities” in the *MPE V System Operator and Resource Management Reference Manual* (32033-90005).

## SHOWLOGSTATUS

Displays status information about currently opened user logging files assigned to a logging identifier.

### Syntax

```
SHOWLOGSTATUS [logid]
```

### Parameters

*logid* Displays the logging identifier created by the GETLOG command. The default is that the status of all logging identifiers is displayed.

### Use

You may issue this command from a session, job, or in **BREAK**. You may not issue it from a program. Pressing **BREAK** has no effect on it.

### Operation

The SHOWLOGSTATUS command lists the status of currently running logging processes as shown in the example below. To display the status of all open log files, enter:

```
:SHOWLOGSTATUS
```

| LOGID    | CHANGE | AUTO | USERS | STATE    | CUR-RECS | MAX-REC | %USED | CUR-SET |
|----------|--------|------|-------|----------|----------|---------|-------|---------|
| DUMMYLOG | NO     | NO   | 4     | INACTIVE | 100      | 1000    | 10%   | 1       |
| TAPELOG  | YES    |      | 1     | INACTIVE | 5738     |         |       | 1       |
| DISCLOG  | YES    | YES  | 2     | INACTIVE | 500      | 1000    | 50%   | 2       |

The meanings of each column are explained below.

| Column | Meaning                                                                                                                                                                                                                                                          |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LOGIN  | The name of the logging process.                                                                                                                                                                                                                                 |
| CHANGE | Whether the CHANGELOG command is permitted, that is, whether the name of the first logging file ends in "001".                                                                                                                                                   |
| AUTO   | Whether an automatic CHANGELOG has been enabled, that is, whether the AUTO parameter has been specified for the ALTLOG or GETLOG commands.                                                                                                                       |
| USERS  | The number of users accessing the logging file.                                                                                                                                                                                                                  |
| STATE  | ACTIVE, INACTIVE, INITIALIZING, or RECOVERING. INACTIVE is displayed when a process is waiting for information from user processes that involve intrinsics. INITIALIZING starts the log process. RECOVERING is displayed immediately after a WARMSTART recovery. |



## **SHOWLOGSTATUS**

| <b>Column</b> | <b>Meaning</b>                           |
|---------------|------------------------------------------|
| CUR-RECS      | The number of records in the log file.   |
| MAX-REC       | The maximum number of records permitted. |
| %USED         | The percentage of the maximum used.      |
| CUR-SET       | The current set number.                  |

### **Additional Discussion**

*MPE V System Operation and Resource Management Reference Manual (32002-90005).*

## SHOWME

Reports the status of your job or session.

### Syntax

```
SHOWME
```

### Parameters

None.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** aborts execution.

### Operation

The **SHOWME** command let you check on your job or session. To display the status of the current job or session, enter:

```
:SHOWME
```

The system responds with the following:

```
USER: #S485,MGR.DSUSER,PUB (NOT IN BREAK)
RELEASE: G.03.08 MPE V HP32033 G.03.08 USER VERSION: G.03.08
CURRENT: MON, AUG 14, 1989 11:09 AM
LOGON: MON, AUG 14, 1989, 11:08 AM
CPU SECONDS: 3 CONNECT MINUTES: 1
$STDIN LDEV: 88 $STDLIST LDEV: 88
```

The **SHOWME** command reports information pertaining to your job or session. Using the example above, the following information is given:

| Item           | Meaning                                                                                                                                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| #S485          | This is the session number. It may also be a job number.                                                                                                                                                                                                                                 |
| (NOT IN BREAK) | An (IN PROGRAM), (IN BREAK), or (NOT IN BREAK) message indicates whether <b>SHOWME</b> was executed programmatically, in <b>BREAK</b> , or directly as an MPE V/E command.                                                                                                               |
| HP32033G.00.00 | This is the user-defined version of the software that you are running. You can modify this number by using the <b>SYSDUMP</b> command. In this example, <b>G</b> is the user-defined version, <b>03</b> is the user-defined update number, and <b>04</b> is the user-defined fix number. |

## SHOWME

| Item           | Meaning                                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (BASE G.00.00) | This number is the base MPE V/E <i>version.update.fix</i> numbers. It is the official Hewlett-Packard version number of the operating system software and cannot be modified by SYSDUMP. |
| CURRENT        | Shows the current time and date.                                                                                                                                                         |
| LOGON          | Shows the logon time.                                                                                                                                                                    |
| CPU SECONDS    | Shows the CPU time used by this job or session.                                                                                                                                          |

---

|                                                                                                  |                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note</b><br> | <b>SHOWME</b> calculates CPU usage by adding the local CPU usage of the current process to the accumulated total of all terminated processes. The CPU usage listed for a programmatic <b>SHOWME</b> , therefore, would rarely agree with that for a <b>SHOWME</b> executed during <b>BREAK</b> . |
|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

|                   |                                                                            |
|-------------------|----------------------------------------------------------------------------|
| CONNECT MINUTES   | The amount of time the job or session has been connected.                  |
| \$STDIN LDEV      | The logical device number of the job or session's standard input device.   |
| \$STDLIST LDEV    | The standard list device number.                                           |
| (WELCOME TO SYS1) | Shows the current welcome message, if the system operator has entered one. |

## SHOWOUT

Displays the status of output device files.

### Syntax

$$\text{SHOWOUT} \left\{ \begin{array}{l} \#nnn \\ \text{STATUS} \\ \text{SP} \\ \text{item} [ ; \text{item} [ ; \dots ] ] \end{array} \right\}$$

### Parameters

|               |                                                                                                                                                                                                                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>#Onnn</i>  | Identifies a particular output device file for which information is to be displayed. The information is displayed in Type I format, described under “Operation”. The default is a display of status information for all output device files used by the logon job or session.                                                        |
| <b>STATUS</b> | Summarizes the status information for all current output device files. The information is displayed in Type II format, described under “Operation”. The default is a display of status information for all output device files used by the logon job or session.                                                                     |
| <b>SP</b>     | Displays the status information for currently spooled output device files associated with the logon job or session. The information is displayed in Type I followed by Type II format, described under “Operation”. The default is a display of status the information for all output device files used by the logon job or session. |
| <i>item</i>   | Displays the status of all current output device files as identified. If information for only one device file is displayed, output is in Type I format; if information for more than one device file is displayed, output is in Type I followed by Type II format. (Refer to “Operation”.) The syntax appears below.                 |

## SHOWOUT

### Syntax for *item*

$$\left[ \text{DEV} = \left\{ \begin{array}{l} ldev \\ classname \end{array} \right\} \right] \left[ \text{JOB} = \left\{ \begin{array}{l} @J \\ @S \\ @ \\ \left[ \# \right] Jnnn \\ \left[ \# \right] Snnn \end{array} \right\} \right] \left[ \left\{ \begin{array}{l} \text{ACTIVE} \\ \text{OPENED} \\ \text{LOCKED} \\ \text{READY} \left[ \begin{array}{l} ,N \\ ,D \end{array} \right] \end{array} \right\} \right]$$

### Parameters for *item*

*ldev* or *classname* Displays the status of output device files. The *ldev* parameter displays the files residing on the device identified by the logical device number. The *classname* parameter displays the status of the output device files residing on all devices in a class name.

**JOB=** Displays the status of output device files using one of the following options:

- @J** Displays the status of output device files for all jobs.
- @S** Displays the status of output device files for all sessions.
- @** Displays the output device files for all jobs and sessions.
- [#] Jnnn** Displays all output device files for a specified job.
- [#] Snnn** Displays the status of all output device files for a specified session.

**ACTIVE, OPENED, READY, or LOCKED** Displays the status of all output files in a specified state. An ACTIVE file is one that is currently being produced on your printer, plotter, or punch card. Only one output spoolfile can be active at any one time. OPENED files are those being accessed by a program. A spool file is opened when a spooler process is writing the file to disk; during that time, however, the file is not ready to be printed, plotted or punched. READY files are completely spooled and ready to be output. A LOCKED file is READY but cannot be accessed until the system relinquishes its exclusive use of the file.

READY files may include one of the following:

- N** Displays the status of nondeferred READY device files.
- D** Displays the status of deferred READY device files.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** aborts execution.

## Operation

The SHOWOUT command displays the status information for one or more currently defined output device files. The information reflects the status at the time the command is entered and always appears on the standard list device. Two types of spooling queues are maintained in MPE V/E: one output queue for each logical device configured on the system, and one additional queue for all device classes. Within each queue, files are linked according to the following parameters and listed in descending order of importance: output priority, device class, and rank. If the priorities are equal, the spooler alternates between queues.

The output may appear in one of two possible formats, or in a combination of the two formats; for example:

Type I:

| DEV/CL | DFID    | JOBNUM | FNAME     | STATE | FRM | SPACE | RANK | PRI | #C |
|--------|---------|--------|-----------|-------|-----|-------|------|-----|----|
| EPOC   | #011470 | #J242  | \$STDLIST | READY |     | 36    | D    | 1   | 1  |

OUTFENCE = 6

Type II:

19 FILES

0 ACTIVE  
 2 READY; INCLUDING 2 SPOOFLES, 2 DEFERRED  
 17 OPENED; INCLUDING 1 SPOOFLE  
 0 LOCKED; INCLUDING 0 SPOOFLES  
 3 SPOOFLES: 1572 SECTORS

OUTFENCE = 6

OUTFENCE = 2 FOR LDEV 13 \*\* LDEV 13 has been assigned an outfence of 2 \*\*

| Column | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------------------------------------------------------------------------------------------|--------|---------------------------------------------------------|--------|------------------------------------------------------------------------------------------------------------------------------------|-------|------------------------------------------------------|
| DEV/CL | Logical device number or device class name of the device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |
| DFID   | Device file identification, in the form #Onnn.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |
| JOBNUM | The number ( <i>jsnum</i> ) of the job or session using the device file not used for READY or ACTIVE data. Otherwise, the job/session name appears online following the standard device information.                                                                                                                                                                                                                                                                                                                                                                            |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |
| FNAME  | File name assigned to the device file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |
| STATE  | The status, indicated by one of the following subparameters: <table border="0"> <tr> <td>ACTIVE</td><td>The spooler device file on disk is actually being written to a printer, plotter, or card punch.</td></tr> <tr> <td>LOCKED</td><td>READY, but the system has exclusive access to the file.</td></tr> <tr> <td>OPENED</td><td>The device file on disk is being accessed by a program. If the device file is spooled, a program is currently writing to the disk.</td></tr> <tr> <td>READY</td><td>The spooled device file on disk is ready for output.</td></tr> </table> | ACTIVE | The spooler device file on disk is actually being written to a printer, plotter, or card punch. | LOCKED | READY, but the system has exclusive access to the file. | OPENED | The device file on disk is being accessed by a program. If the device file is spooled, a program is currently writing to the disk. | READY | The spooled device file on disk is ready for output. |
| ACTIVE | The spooler device file on disk is actually being written to a printer, plotter, or card punch.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |
| LOCKED | READY, but the system has exclusive access to the file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |
| OPENED | The device file on disk is being accessed by a program. If the device file is spooled, a program is currently writing to the disk.                                                                                                                                                                                                                                                                                                                                                                                                                                              |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |
| READY  | The spooled device file on disk is ready for output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |        |                                                                                                 |        |                                                         |        |                                                                                                                                    |       |                                                      |

## SHOWOUT

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FRM   | The forms message indicator, F, appears only if a forms alignment message applies to this device file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| SPACE | The approximate disk space currently used in sectors. This applies only to spooled output device files.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| RANK  | <p>Indicates the order in which the file is entered into the system with respect to other files of the same output priority and <i>classname</i> or <i>ldev</i>. A time stamp activated by the FCLOSE command determines the file's rank.</p> <p>Rank is not a direct indication of when a file becomes active. Along with other factors, including output priority, class, and the availability of the specified device, the rank helps to determine the order in which READY output device files will be written to the output device file (made ACTIVE). Because of the complex nature of the spooling environment and the fact that it is constantly changing, it is impossible to predict, on the basis of rank or any other factor, exactly when a file will become active. A "D" following RANK indicates a deferred file. This applies only to spooled device files. A file can be deferred if its priority is less than or equal to the system outfence or to the outfence of a specific device.</p> <p>The RANK column of a SHOWOUT listing for a single-printer system may be confusing because it is possible for two device files to have the same rank and output priority. The reason for this is that two types of spooling queues are maintained in MPE V/E: a device class queue and <i>ldev</i> queues (one for each logical device configured on the system). The rank of a device file indicates when the file entered the queue to which the user or program directed it. For example, DEV=LP sends the file to a device class queue, and LDEV=6 sends it to a specific logical device.</p> <p>As the spooler searches for files to print, it compares candidates from each of the queues and follows a complex procedure to determine which file to print next. The rank of the file helps determine when a file is selected from among the files in its own queue, but it is not a factor in the selection of files from separate queues. So, a system with only one printer and one spooler process for that printer may legitimately produce a SHOWOUT listing with two device files of the same rank, although at first sight it may seem to be a duplication.</p> |
| PRI   | The output priority requested by user or as adjusted by the system operator for spooled device files only. A priority of 1 is lowest, and 13 is highest.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| #C    | Number of copies needed, for spooled device files only.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

## Examples

To display the total number of output device files currently existing, the number of those that are spooled, and the states that they are in, enter:

```
:SHOWOUT STATUS
11 FILES:
 1 ACTIVE
 1 READY; INCL 1 SPOOFLES, 0 DEFERRED
 9 OPENED; INCL 1 SPOOFLES
 0 LOCKED; INCL 0 SPOOFLES
 3 SPOOFLES: 7212 SECTORS
OUTFENCE= 2
:
```

To list information about an individual output device file, you may specify its device file identifier (DFID) in the SHOWOUT command:

```
:SHOWOUT #011470

DEV/CL DFID JOBNUM FNAME STATE FRM SPACE RANK PRI #C
EPOC #011470 #J242 $STDLIST READY 36 D 1 1

OUTFENCE = 6
:
```

You can also request information about a specific output device file, device number or device class name of the device for which the file is destined in the SHOWOUT command:

```
:SHOWOUT DEV=43

DEV/CL DFID JOBNUM FNAME STATE FRM SPACE RANK PRI #C
43 #089 #S37 $STDLIST OPENED

OUTFENCE= 2
:
```



---

## SHOWQ

Displays process scheduling data and the contents of each subqueue.

### Syntax

SHOWQ

### Parameters

None.

### Use

You may issue this command from a session, job, in **BREAK**. You may not issue it from a program. Pressing **BREAK** aborts execution. You require system supervisor (OP) capability to use it.

### Operation

The **SHOWQ** command let you display process scheduling and subqueue information. The data is displayed in three major columns: **DORMANT**, **WAITING**, and **RUNNING**. **RUNNING** processes are these that currently require the CPU in order to continue, or that require it in the immediate future. CPU time is automatically allocated to the highest-priority process that is ready to run.

The processes listed in **DORMANT** and **WAITING** columns are waiting for longer-term events. **DORMANT** processes must wait a shorter amount of time than **WAITING** processes, for example, until a code segment moves into memory. **WAITING** processes wait for a longer event, such as an I/O completion. They contend for the CPU once those events have occurred.

On occasion, a process appears in more than one column. This indicates that the process was being moved one column to the right when you executed **SHOWQ**.

Below is a sample of the output listing produced by the **SHOWQ** command. The symbols that may appear in such a listing are explained in the remainder of the discussion.

| DORMANT |     |        | WAITING |     |        | RUNNING |      |        |
|---------|-----|--------|---------|-----|--------|---------|------|--------|
| Q       | PIN | JOBNUM | Q       | PIN | JOBNUM | Q       | PIN  | JOBNUM |
| L       | 1   |        |         |     |        | C       | M163 | #S263  |
| L       | 2   |        |         |     |        | C       | U215 | #S256  |
| L       | 3   |        |         |     |        |         |      |        |
| L       | 4   |        |         |     |        |         |      |        |
| D       | U29 | #J30   |         |     |        |         |      |        |
| C       | M37 | #S234  |         |     |        |         |      |        |
| C       | M55 | #S248  |         |     |        |         |      |        |

Each entry in the three columns displays the following information for a single process:

$$\left\{ \begin{array}{c} L \\ C \\ D \\ E \end{array} \right\} \left[ \begin{array}{c} M \\ U \end{array} \right] \text{pin} \left[ \begin{array}{c} \#Jnnn \\ \#Snnn \end{array} \right]$$

where:

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

The process identification number (PIN) may appear with or without an M or U label. Processes without an M or U label are system processes.

In addition, SHOWQ prints the scheduling parameters currently in effect along with the minimum memory manager algorithm cycle value (*minclockcycle*).

```
CQ MINQUANTUM=0, MAXQUANTUM=300, BASEPRI=152, LIMITPRI=200
DQ MINQUANTUM=1000, MAXQUANTUM=1000, BASEPRI=202, LIMITPRI=238
EQ MINQUANTUM=1000, MAXQUANTUM=1000, BASEPRI=153, LIMITPRI=154
MINIMUM CLOCK CYCLE=1000
```

## Example

To display the queues of processes within MPE V/E, enter:

```
:SHOWQ
```

The system displays a listing of DORMANT, WAITING, and RUNNING processes similar to the one explained in the preceding "Operation" section.

---

## SHOWTIME

Prints current time and date.

### Syntax

```
SHOWTIME
```

### Parameters

None.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on it.

### Operation

The **SHOWTIME** command lets you print the current time and date, as indicated by the system clock.

### Example

To display the time and date, enter:

```
:SHOWTIME
MON, AUG 14, 1989, 8:47 AM
```

---

## **=SHUTDOWN**

Initiates a shutdown of MPE V/E.

### **Syntax**

**=SHUTDOWN**

### **Parameters**

None.

### **Use**

You may issue this command from a session, job, in **BREAK**, but not from a program. Pressing **BREAK** has no effect on this command. You can issue it only at the console.

### **Operation**

The **=SHUTDOWN** command performs an implicit **=LOGOFF** of all sessions, including the session logged on at the system console. All system processes are stopped in an orderly fashion. This includes the completion of all pending system activity and any processing necessary to ensure that the integrity of all system tables and directories is maintained. Once these procedures are complete, "SHUT" is displayed on the console, the CPU halts, and 'console interrupt' (**CTRL A**) is ineffective.

Device configuration changes that were made after the preceding load (**UP**, **DOWN**, **GIVE**, **TAKE**, **ACCEPT**, **REFUSE** and spooling commands) are not retained. Configuration changes made during the cold load are permanently recorded and retained until the next tape cold load. Newly assigned or released global resource identification numbers (RINs) are permanently recorded.

All communication lines must be closed before you issue a **=SHUTDOWN** command. Otherwise, you may have to carry out a manual halt of the system. Data is lost if a transmission is in progress when the halt is performed. Also, if DSN/DS lines were left open when you issued the **=SHUTDOWN** command, lines to the remote system remain open and any remote sessions are effectively "hung". In this case, the remote system's operator may need to issue **ABORTIO** commands for the hung sessions and then abort them.

Spooled devices stop operating immediately upon receiving a **=SHUTDOWN** command. A **WARMSTART** keeps spool files so that they may be printed when the system is returned online.

## **=SHUTDOWN**

### **Example**

To shut the system down, first issue a warning to all users to allow them time to log off, and then execute =SHUTDOWN as shown below:

```
:WARN @;SYSTEM WILL SHUTDOWN IN FIVE MINUTES. PLS LOG OFF.
```

```
CTRL A ** Press A and CTRL simultaneously **
=SHUTDOWN
10:49/#S40/25/LOGOFF
10:49/20/ALL JOBS LOGGED-OFF
```

## SHUTQ

Closes the spool queue for the specified logical device or device class.

### Syntax

$$\text{SHUTQ} \left\{ \begin{array}{l} ldev \\ devclass \end{array} \right\}$$

### Parameters

*ldev*                      The logical device number of the spooled device.

*devclass*                The device class name of the spooled device.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You may issue it only from the console unless it is distributed to users with the **ALLOW** command.

### Operation

The **SHUTQ** command closes the spool queue for a logical device or device class configured in the system. The spooler process, however, does not need to be running for the device. If the spooler process is running, it is unaffected by your shutting the queue.

This command also serves as an option to the **STARTSPOOL** command, which is documented in this chapter.

### Example

To shut the queue for device class **LP**, enter:

:SHUTQ LP

---

## SPEED

Changes the operating speed of the terminal.

### Syntax

**SPEED** *newinspeed, newoutspeed*

### Parameters

- newinspeed*      The new input speed in characters per second (CPS). For Series 39/40/42/44/58/64/68/70 the speed must be 15, 60, 120, 240, 480, or 960. Series 39/40/42/44/48 also supports 10 CPS. If you are using the ATP driver, the speed 1020 is also supported. The *newinspeed* must be the same as *newoutspeed*.
- newoutspeed*    The new output speed in characters per second. For Series 39/40/42/44/58/64/68/70 the speed must be 15, 60, 120, 480, or 960. Series 39/40/42/44/48 also supports 10 CPS. If you are using the ATP driver, the speed 1020 is also supported. The *newoutspeed* must be the same as *newinspeed*.

### Use

You may issue this command from a session, in **BREAK**, or from a program, but not from a job. Pressing **BREAK** has no effect on this command.

### Operation

MPE V/E automatically senses the input/output speed of a terminal when you logon at that terminal. If your terminal has speed adjustment controls, you can change the input and output speeds after logon with the **SPEED** command. This command is not valid for terminals that operate at only one speed. When you enter the command, MPE V/E displays the following message at the old output speed:

CHANGE SPEED AND INPUT "MPE":

Manually change the speed control on the terminal and verify the new speed by entering:

:**MPE** **RETURN**

If the characters "MPE" cannot be verified, the system assumes that the terminal is to continue at the old speed. (To continue, you must reset the terminal control to the old speed.) On Hewlett-Packard terminals, the baud rate is characters per second (CPS) multiplied by 10. When you select the baud rate at which you choose to operate, you must, therefore, divide the rate by 10, and enter that value with the **SPEED** command.

You can also change the terminal speed programmatically by using the **FCONTROL** intrinsic. Refer to the *MPE V Intrinsic Reference Manual* (32033-90007).

Refer to appendix A, "Terminals Supported by MPE V", for a list of the terminal types and the speeds that they support.

## Examples

To change the input and output speeds to 240 CPS (2400 baud), enter:

:SPEED 240,240

CHANGE SPEED AND INPUT "MPE":

{

Manually change the speed and type in MPE. The "{" is a random character.

{MPE RETURN

## Additional Discussion

*Fundamental Communications Handbook* (5957-4634)





---

## SPL

Compiles an SPL program. SPL is not part of the HP 3000 fundamental operating system and must be purchased separately.

### Syntax

```
SPL [textfile] [, [uslfile] [, [listfile] [, [masterfile] [, newfile]]]] [;INFO=quotedstring]
```

### Parameters

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>textfile</i>   | Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is SPLTEXT. The default is \$STDIN.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <i>uslfile</i>    | <p>Actual file designator of the user subprogram library (USL) file to which the object code is written. This can be any binary output file created with a file code of USL or 1024. Its formal file designator is SPLUSL. If the <i>uslfile</i> parameter is omitted, the object code is saved to the temporary file \$OLDPASS. If the <i>uslfile</i> parameter is entered, it indicates that the file was created in one of four ways:</p> <ul style="list-style-type: none"><li>■ By using the SAVE command to save the default USL file created during a previous compilation.</li><li>■ By building the USL with the Segmenter -BUILDUSL command. Refer to the <i>MPE Segmenter Reference Manual</i> (30000-90011).</li><li>■ By creating a new USL file with the BUILD command and specifying a file code of USL or 1024.</li><li>■ By having the statement \$CONTROL USLINIT in your program.</li></ul> |
| <i>listfile</i>   | Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is SPLLIST. The default is \$STDLIST.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <i>masterfile</i> | Actual file designator of the master file with which <i>textfile</i> is merged to produce a composite source. This can be any ASCII input file. The formal file designator is SPLMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>newfile</i>    | Actual file designator of the file created by merging <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. Formal designator is SPLNEW. The default is that no file is written.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

---

### Note



The formal file designators used in this command (SPLTEXT, SPLUSL, SPLLIST, SPLMAST, and SPLNEW) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the FILE command.

---

*quotedstring* A sequence of ASCII characters bounded by a pair of single quotation marks or by double quotation marks. If you want a quotation to appear within *quotedstring*, the quotation and its quotation marks must also be bounded by quotation marks. For example, to insert “and” into a *quotedstring*, it must appear as ””and””. Similarly, 'and' must appear as ”and”. The maximum length of the string, including delimiters, is 255 characters. Refer to “Operation”.

For SPL to recognize *quotedstring*, a dollar sign (\$) must follow the quotation marks at the beginning of the *quotedstring*. This feature is used to specify compiler options which appear at the beginning of the source listing. For more information, refer to the **RUN** command in this chapter.

## Use

You may issue this command from a session or a job, but not in **BREAK** or from a program. Pressing **BREAK** suspends execution.

## Operation

The SPL command compiles an SPL program into a user subprogram library (USL) file on disk. If *textfile* is not specified, MPE V/E expects the source program to be entered from your standard input device. If *listfile* is not specified, the program output is sent to your standard list device.

## Examples

The following example compiles an SPL program entered from your standard input device into an object program in the USL file \$OLDPASS, and writes the listing to your standard list device:

```
:SPL
```

The next example compiles an SPL program contained into the disk file **SOURCE** and stores the object code into the USL file **OBJECT**. The program listing is sent to the disk file **LISTFL**:

```
:SPL SOURCE,OBJECT,LISTFL
:SAVE OBJECT
```

## Additional Discussion

*Systems Programming Language Reference Manual (30000-90024)*

---

## SPLGO

Compiles, prepares, and executes an SPL program. SPL is not part of the HP 3000 fundamental operating system and must be purchased separately.

### Syntax

```
SPLGO [textfile] [, [listfile] [, [masterfile] [, newfile]]] [;INFO=quotedstring]
```

### Parameters

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>textfile</i>   | Actual file designator of the input file from which the source program is read. This can be any ASCII input file. The formal file designator is SPLTEXT. The default is \$STDIN.                                                                                                                                                                                                                                                                                     |
| <i>listfile</i>   | Actual file designator of the file to which the program listing is written. This can be any ASCII output file. The formal file designator is SPLLIST. The default is \$STDLIST.                                                                                                                                                                                                                                                                                      |
| <i>masterfile</i> | Actual file designator of the master file that is merged against <i>textfile</i> to produce a composite source. This can be any ASCII input file. Formal file designator is SPLMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified. If two files being merged have identical line numbers, the lines from <i>textfile</i> or from \$STDIN overwrite those in masterfile. |
| <i>newfile</i>    | Actual file designator of the file produced by merging <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. The formal file designator is SPLNEW. The default is that no file is written.                                                                                                                                                                                                                                                      |

---

### Note



The formal file designators used in this command (SPLTEXT, SPLLIST, SPLMAST, and SPLNEW) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands For Subsystems” under the FILE command.

---

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>quotedstring</i> | <p>A sequence of ASCII characters bounded by a pair of single quotation marks or by double quotation marks. If you want a quotation to appear within <i>quotedstring</i>, the quotation and its quotation marks must also be bounded by quotation marks. For example, to insert “and” into a <i>quotedstring</i>, it must appear as “”and””. Similarly, ‘and’ must appear as “and”. The maximum length of the string, including delimiters, is 255 characters.</p> <p>For SPL to recognize <i>quotedstring</i>, a dollar sign (\$) must follow the quotation marks at the beginning of the <i>quotedstring</i>. This feature is used to specify compiler options which appear in front of the source listing. For more information, refer to the RUN command in this chapter.</p> |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Use

You may issue this command from a session or a job, but not in **BREAK** or from a program. Pressing **BREAK** suspends execution.

## Operation

The **SPLGO** command compiles, prepares, and executes an SPL program. If *textfile* is omitted, MPE V/E expects input from your standard input device. This command creates a temporary user subprogram library (USL) file (**\$NEWPASS**) that you cannot access, and a temporary program file that you can access under the name **\$OLDPASS**. For more information, refer to “System-Defined Files” in the introduction to this manual.

## Examples

To compile, prepare, and execute an SPL program entered from your standard input device, and have the program listing sent to your standard list device, enter:

:SPLGO

To compile, prepare, and execute an SPL program read from the disk file **SOURCE** and send the resulting program listing to the disk file **LISTFL**, enter:

:SPLGO SOURCE,LISTFL

## Additional Discussion

*Systems Programming Language Reference Manual*  
(30000-90024)  
*MPE Segmenter Reference Manual* (30000-90011)

---

## SPLPREP

Compiles and prepares an SPL program. SPL is not part of the HP 3000 fundamental operating system and must be purchased separately.

### Syntax

```
SPLPREP [textfile] [, [progfile] [, [listf] [, [masterf] [, newfile]]]]
[; INFO=quotedstring]
```

### Parameters

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>textfile</i>     | Actual file designator of the input file from which source program is read. This can be any ASCII input file. Formal file designator is SPLTEXT. The default is \$STDIN.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <i>progfile</i>     | Actual file designator of the program file to which the prepared program segments are written. You get the best results by omitting the <i>progfile</i> parameter. When you omit <i>progfile</i> , the Segmenter creates the program file, which resides in the temporary file domain as \$OLDPASS. If you do create your own program file, you must do so in one of two ways: <ul style="list-style-type: none"><li>■ By using the BUILD command, specifying a file code of 1029 or PROG, and a <i>numextents</i> value of 1. This file is then used by the PREP command.</li><li>■ By specifying a nonexistent file in the <i>progfile</i> parameter, in which case a job/session temporary file of the correct size and type is created.</li></ul> |
| <i>listf(ile)</i>   | Actual file designator of the file to which program listing is written. This can be any ASCII output file. Formal designator is SPLLIST. The default is \$STDLIST.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <i>masterf(ile)</i> | Actual file designator of the master file that is merged against <i>textfile</i> to produce composite source. This can be any ASCII input file. The formal file designator is SPLMAST. The default is that the master file is not read; input is read from <i>textfile</i> , or from \$STDIN if <i>textfile</i> is not specified. If two files being merged have identical line numbers, the lines from <i>textfile</i> or from \$STDIN overwrite those in <i>masterfile</i> .                                                                                                                                                                                                                                                                        |
| <i>newfile</i>      | Actual file designator of the file produced by merging <i>textfile</i> and <i>masterfile</i> . This can be any ASCII output file. The formal file designator is SPLNEW. The default is that no file written.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

---

### Note



The formal file designators used in this command (SPLTEXT, SPLLIST, SPLMAST, and SPLNEW) cannot be backreferenced as actual file designators in the command parameter list. For further information, refer to the discussion on “Implicit FILE Commands for Subsystems” under the FILE command.

---

*quotedstring* A sequence of ASCII characters bounded by a pair of single quotation marks (apostrophes) or by double quotation marks. If you want a quotation to appear within *quotedstring*, the quotation and its quotation marks must also be bounded by quotation marks. For example, to insert “and” into a *quotedstring*, it must appear as ””and””. Similarly, ‘and’ must appear as ’and’. The maximum length of the string, including delimiters, is 255 characters. Refer to “Operation”.

For SPL to recognize *quotedstring*, a dollar sign (\$) must follow the quotation marks at the beginning of the *quotedstring*. This feature is used to specify compiler options which appear at the beginning of the source listing. For more information, refer to the RUN command in this chapter.

## Use

You may issue this command from a session or a job, but not in BREAK or from a program. Pressing **BREAK** suspends execution.

## Operation

The SPLPREP command compiles and prepares an SPL program into a program file on disk. If *textfile* is not specified, MPE V/E expects you to enter your source program from your standard input device. If you do not specify *listfile*, your program output is sent to your standard list device.

The user subprogram library (USL) file created during compilation, \$OLDPASS, is a temporary file passed directly to the Segmenter. It can be accessed only if you do not use the default for *progfile*. This is because the Segmenter also uses \$OLDPASS to store the prepared program segments, overwriting the USL file of the same name.

## Examples

To compile and prepare an SPL program entered from your standard input device and send the output to your standard list device, enter:

```
:SPLPREP
```

The following example compiles and prepares an SPL source program from the disk file SFILE into the program file MYPROG. The program listing is sent to your standard list device:

```
:SPLPREP SFILE,MYPROG
```

## **SPLPREP**

In the next example, the first positional parameter is omitted. This indicates to MPE V/E that you intend to enter the source text from your standard input device. The object code is stored in the default USL file \$OLDPASS, and the prepared program segments are stored in FILEZ.OLDPASS is then saved in the permanent file domain under the new name NUSL.

```
:SPLPREP,FILEZ
:SAVE $OLDPASS, NUSL
```

## **Additional Discussion**

*System Programming Language Reference Manual (30000-90024)*

*MPE Segmenter Reference Manual (30000-90011)*

---

## STARTCACHE

Enables caching on a single disk. Disk caching software is not included with the fundamental operating system and must be purchased separately.

### Syntax

`STARTCACHE ldev`

### Parameter

*ldev*                The logical device number of the disk to be used for caching.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You require system supervisor (OP) or system manager (SM) capability to use it.

### Operation

The system supervisor uses the **STARTCACHE** command to enable caching on a single disk. When the system is first started, caching is disabled by default. You must invoke the command separately for each disk to be cached.

### Example

To enable caching on *ldev* 3, enter:

```
:STARTCACHE 3
CACHING STARTED ON LDEV 3.
```

---

### Note

If caching is not installed on your system, an error message is displayed:



CIERR 4417 DISC CACHING NOT ENABLED ON THIS SYSTEM

---



---

## STARTSESS

Creates a session on the specified device, provided that the command's user has programmatic sessions (PS) capability.

### Syntax

```
STARTSESS ldev[sessionname,] user[/userpass].acct [/acctpass]

[, group[/grouppass]] [;TERM={ termtype }] [;TIME=cpusecs] [;PRI= { BS
CS
DS
ES }]

[{ ;INPRI=inputpriority }] [;NOWAIT]
```

### Parameters

*ldev*                The logical device number of the target terminal. This terminal must be a real physical device and cannot be a virtual terminal or a DS pseudo-terminal. The terminal must be configured as **TYPE 16** and as **SUBTYPE 0** or **4**.

### Note



You must specify all required passwords in the command invocation; **STARTSESS** does not prompt the caller or the *ldev* for passwords. If you omit a password where one is required, the command fails.

|                    |                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>sessionname</i> | Arbitrary name used in conjunction with the <i>user</i> and <i>acct</i> parameters to form a fully qualified session identity. The name may contain from one to eight alphanumeric characters, beginning with an alphabetic character. The default is that no session name is assigned.                                                                                       |
| <i>user</i>        | User name, established by the account manager, that allows you to logon under this account. The name may contain from one to eight alphanumeric characters, beginning with an alphabetic character.                                                                                                                                                                           |
| <i>userpass</i>    | User password, optionally assigned by the account manager. The password may contain from one to eight alphanumeric characters, beginning with an alphabetic character.                                                                                                                                                                                                        |
| <i>acct</i>        | Account name established by the system manager. The name may contain from one to eight alphanumeric characters, beginning with an alphabetic character. A period (.) must precede the <i>acct</i> parameter.                                                                                                                                                                  |
| <i>acctpass</i>    | Account password, optionally assigned by the system manager. The password may contain from one to eight alphanumeric characters, beginning with an alphabetic character. A slash (/) must precede the <i>acctpass</i> parameter.                                                                                                                                              |
| <i>group</i>       | Group name to be used for the local file domain and the CPU time charges established by the account manager. The name may contain from one to eight alphanumeric characters, beginning with an alphabetic character. The default is the specified users home group if you are assigned one by the account manager. The parameter is required if a home group is not assigned. |

|                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>grouppass</i>                      | The <i>grouppass</i> parameter is not needed when the user logs on under the user's home group, even if a password has been established. The <i>grouppass</i> is needed when the user logs on under any other group for which a password has been established. A slash (/) must precede the parameter <i>grouppass</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <i>termtype</i> or<br><i>termname</i> | <p>Determines terminal-type characteristics. The value of the <i>termtype</i> parameter determines the type of terminal used for input. MPE V/E uses this parameter to determine device-dependent characteristics such as delay factors for carriage returns. The value must be a number from 0 to 22. The default value for <i>termtype</i> is assigned by the system supervisor during system configuration. This parameter is required to ensure correct listings if your terminal is not the default <i>termtype</i>.</p> <p>The <i>termname</i> parameter is the name of the file containing the desired terminal-type characteristics. The file must not have a lockword or reside on a Private Volume.</p> <p>Users of the workstation configurator are allowed to create terminal-type files. The proper and efficient operation of a specific device by a user-created terminal-type is the responsibility of the user. The workstation configurator utility allows the user to specify the following characteristics of the terminal: data flow control, block mode, READ trigger, special characteristics, echo, line feed, parity, and printer control.</p> <p>If group and/or account names are omitted, the proposed logon group and/or account name is substituted. For further information on the terminal-type characteristics, refer to the <i>Communications Handbook</i> (30000-90105) and the <i>Workstation Configurator Reference Manual</i> (30239-90001). Refer also to appendix A, "Terminals Supported by MPE V/E".</p> |
| <i>cpusecs</i>                        | Maximum CPU time that a session must use, entered in seconds. When the limit is reached, the session is aborted. It must be a value from 1 to 32767, provided that it does not exceed any limit imposed by the system or account manager. To specify no limit, enter a question mark (?), or UNLIM, or omit the parameter. The default is no limit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| BS,CS,DS,or ES                        | The execution priority queue that the Command Interpreter uses for your session, and also the default priority for all programs executed within the session. BS is highest priority, ES is lowest. The default is CS. If you specify a priority that exceeds the highest permitted for your account or username by the system, MPE V/E assigns the highest priority possible below BS. DS and ES are intended primarily for batch jobs; their use for sessions is generally discouraged.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

**Note**

Use care when assigning the BS queue because processes in this priority class must lock out other processes. For information on the guidelines for these priority queues, refer to the TUNE command in this chapter.

## STARTSESS

*inputpriority* or HIPRI Determines the input priority of the job or session. The *inputpriority* option is the relative input priority used in checking against access restrictions imposed by the jobfence. The *inputpriority* option takes effect at logon time and must be from 1 (lowest priority) to 13 (highest priority). If you supply a value less than or equal to current jobfence set by system operator, session is denied access. The default is 8.

The HIPRI option is used for two different purposes when logging on. It can be used to override the system jobfence or it can be used to override the session limit. When using the HIPRI option to override the jobfence, the system first checks to see if you have system manager (SM) or system operator (OP) capability. The user who has either of these capabilities, is logged on and the INPRI defaults to the system's jobfence and execution limit. If you do not have either of these capabilities, the system attempts to log you on using INPRI=13 and succeeds if the jobfence is 12 or less, and the session limit is not exceeded. In attempting to override the session limit (to logon after the maximum number of sessions set by the system operator has been reached), you can specify HIPRI, but in this case you must have either SM or OP capability. The system does not override the session limit automatically. Use of the HIPRI option without SM or OP capability causes the following warning to be displayed:

MUST HAVE 'SM' OR 'OP' CAP. TO SPECIFY HIPRI,  
MAXIMUM INPRI OF 13 IS USED (CIWARN 1460)

NOWAIT Request that the session starts executing immediately without waiting for a **RETURN** on the terminal. If this parameter is specified and the target terminal is the system console, system manager (SM) capability is required.

### Use

This command is available from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You require programmatic sessions (PS) capability to use it.

### Operation

Use the **STARTSESS** command to create a session at any terminal on the system. The effect is the same as logging on at the target terminal.

---

#### Note



1. The target terminal must be available; no other user must be logged on, and the terminal must be turned on.
  2. No speed sensing is done for the target terminal, so it must be set at the configured baud rate.
  3. When the session is created, nothing is printed to the target terminal until its carriage return key is pressed, unless you specified **NOWAIT**.
-

## **Passwords**

If you are issuing **STARTSESS** for an account, user, or group that has a password, you must provide that password when you issue the command. You can either embed the passwords required in the logon string using the slash mark (/), or you can let MPE V/E prompt you for any required passwords.

The password prompt is similar to the one for the **HELLO** command; MPE V/E prompts you for the password, then turns off echoing so the password is not printed. If you enter the password incorrectly, the prompt reappears and you have two more chances to enter the password correctly. After the third incorrect entry, the **STARTSESS** command aborts.

## **Example**

To start a session named **ALPHA**, with the username **MAC**, accountname **TECHPUBS**, groupname **XGROUP**, grouppass **XPASS**, on logical device **#21**, enter:

```
STARTSESS 21;ALPHA,MAC.TECHPUBS,XGROUP/XPASS
PROGRAMMATIC LOGON FOR SESSION #S7258
```

---

## STARTSPOOL

Initiates the spooler process for a device.

### Syntax

$$\text{STARTSPOOL } \left\{ \begin{array}{l} ldev [ ; \text{SHUTQ} ] \\ devclass \end{array} \right\}$$

### Parameters

|                 |                                                                                                                                                                                                |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ldev</i>     | The logical device number of a spooled device. When the spooler gains control of the specified device, it controls spooling to it as well as to all device classes which reference the device. |
| <i>devclass</i> | The device class specified in the I/O configuration. Only this device class becomes spooled; it does not affect other device classes or any devices in the class.                              |
| SHUTQ           | The spooler prints files waiting in the queue for device <i>ldev</i> , but prevents the creation of new spool files. The SHUTQ parameter is valid for <i>ldev</i> only.                        |

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You may issue it only at the console unless it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

### Operation

To start the spooling process for a specified device, and for any and all device classes associated with it, issue the **STARTSPOOL** command with the *ldev* parameter. When *devclass* is used with **STARTSPOOL**, only the specified device class is controlled by the spooler. The logical device itself is not controlled, unless a **STARTSPOOL** has also been issued for the corresponding *ldev*.

The HP 2680A and HP 2688 laser printers operate only as spooled devices. If spooling is enabled only on the laser printer's *ldev*, and spooling stops as a result of an I/O error, no new spool files are created. To avoid this, issue the **STARTSPOOL** command twice for an HP 2680 laser printer: once for the *devclass* associated with the printer, and a second time for the *ldev* assigned to it.

---

### Note



If the *ldev* referenced in a **STARTSPOOL** command is an HP 2894 card reader/punch, the following message appears on the system console:

IS THIS AN INPUT OR OUTPUT SPOOLER (IN/OUT)?

You must then enter the appropriate reply: "IN", if cards are to be read; "OUT", if cards are to be punched.

---

**Examples**

To start spooling all output to logical device #6 and all device classes that reference logical device #6, enter:

:STARTSPOOL 6

To start spooling all output to device class LP, enter:

:STARTSPOOL LP

To start spooling on logical device #6 while preventing the creation of any new spool files, enter:

:STARTSPOOL 6;SHUTQ

**Additional Discussion**

Refer to the discussion on “SPOOLING” in the *MPE System Operation and Resource Reference Manual* (32033-90005).



---

## STOPCACHE

Disables caching on a single disk. Disk caching software is not included in the fundamental operating system, and must be purchased separately.

### Syntax

**STOPCACHE** *ldev*

### Parameters

*ldev*                The logical device number of the disk for which you intend to disable caching.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You require system supervisor (OP) or system manager (SM) capability to use it.

### Operation

The system supervisor uses the **STOPCACHE** command to disable caching on a single disk. If your system uses more than one disk for caching, you must disable each *ldev* with a separate invocation of **STOPCACHE**.

### Example

To disable caching on *ldev* 3, enter:

:STOPCACHE 3

## STOPSPPOOL

Terminates spooling to a specified device or device class.

### Syntax

```
STOPSPPOOL { ldev[; OPENQ] }
 devclass
```

### Parameters

|                 |                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ldev</i>     | The logical device number of a spooled device. The spooler process gives up ownership of the spooled device. If the OPENQ parameter is omitted, the device becomes available only for non-spoiled I/O. When a logical device is assigned to more than one device class, spooling may be restarted for a specific device class by issuing an explicit STARTSPPOOL request for that class. |
| <i>devclass</i> | The device class specified in system I/O configuration. Subsequent I/O directed to this device class does not take place to or from a spool file. I/O goes directly to or from a logical device if one is available within the device class. If none is available, the program is unable to open the file.                                                                               |
| OPENQ           | May be specified with the <i>ldev</i> parameter only. The spooler process leaves the queue open, or opens the queue if it was previously shut. The default is SHUTQ.                                                                                                                                                                                                                     |

### Use

You may issue this command from a session, job, in BREAK, or from a program. Pressing **BREAK** has no effect on this command. You may issue it only from the console unless it is distributed to users with the ALLOW or ASSOCIATE commands.

### Operation

You may use the STOPSPPOOL command to stop spooling for a single logical device, or for all devices assigned a common device class. If you use the *devclass* parameter in a STOPSPPOOL command, you shut the queue for that device class. When you specify *ldev*, however, you may shut the spooling queue or leave it open. The default is SHUTQ.

When spooling is stopped for a particular device without using OPENQ, that device is conversationally referred to as “hot” (a “hot printer”, for example). In this case, one job has exclusive access to that device preventing other programs from accessing it.



## STOPSPPOOL

### Examples

To terminate spooling to logical device number #6 and cause the spooler process to relinquish control of that device, enter:

:STOPSPPOOL 6

Spooling also terminates for any device class that references this device unless a STARTSPPOOL has been issued for a specific device class.

To stop directing output for device class LP to a spool file (provided a STOPSPPOOL 6 has also been issued), enter:

:STOPSPPOOL LP

To terminate spooling on logical device #6 and leave the queue open, enter:

:STOPSPPOOL 6;OPENQ

### Additional Discussion

Refer to the discussion oo “SPOOLING” in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

## STORE

Copies disk files onto magnetic tape or serial disk. Use the **RESTORE** command to copy files back to disk.

### Syntax

```
STORE [{ fileset[-fileset] [,fileset[-fileset]] [, ...] }] [{ ;*storefile }]
 [;!indirectfile]
 [;SHOW[=showparm[,showparm[, ...]]]] [;FILES=maxfiles]
 [;DATE[{ <=accdate }]] [;ONERR[OR]={ QUIT }] [;PURGE]
 [;PROGRESS[=#minutes]] [;STORESET=tapelistspec] [;INTER[leave]]
 [;COPYACD]
```

### Parameters

*fileset* Specifies the set or sets of files to be stored. You must have **READ** and **LOCK** access to files in order to store them. This positional parameter has the form:

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

If you do not specify a group or account, MPE V/E stores files in your logon group and account. The default file name is @, which stores all files in the logon group.

You can use the characters @, #, and ? as wildcard characters to refer to a group of files.. The wildcard characters count toward the eight-character limit for the user, group, account, and file names. These wildcard characters have the following meanings:

- @ Represents zero or more alphanumeric characters.  
Used alone, @ denotes all members of the set.
- # Represents one digit.
- ? Represents one alphanumeric character.

You might use the wildcard characters as shown in the following examples:

- n@ Stores all items beginning with the character "n".
- @n Stores all items endingwith the character "n".
- n@x Stores all items starting with the character "n" and ending with the character "x".
- n##### Stores all items starting with the character "n" followed by seven digits.
- ?n@ Stores all items whose second character is "n".
- n? Stores all two-character items beginning with an "n".
- ?n Stores all two-character items ending with an "n".

## STORE

*-fileset* Name of a file set or subfile set to be excluded. You can have only one excluded subset for each file set, and you cannot specify excluded subsets within excluded subsets. A negative sign (-) must precede this parameter.

*!indirectfile* An ASCII file containing parameters for the STORE command. Any line of the *indirectfile* may contain *filesets* and keywords. An exclamation point (!) must precede this parameter.

*\*storefile* The name of a single tape or serial disk device file on which to store the files. It can be any magnetic tape or serial disk from the output set. Use it only if you are storing files to a single device. Use the STORESET option to specify multiple backup devices. Use a FILE command to name the output file, and an asterisk (\*) to backreference that file in your STORE command.

**SHOW** Produces a list of all files stored. As each file is stored, STORE prints the following information on your terminal: filename, group name, account name, logical device number, file size (in sectors) and file code. The STORE program also displays error messages on your terminal. If a particular file is not stored for any reason, you may elect to abort the store process. STORE sends the listing is to \$STDLIST (designator: SYSLIST) unless you issue a FILE command to send the listing to some other output device. If you do not use the SHOW parameter, STORE only lists the names of the files not stored and the number of files stored and not stored.

*showparm* May be SHORT, LONG, DATES, SECURITY, or OFFLINE. You cannot specify both LONG and SHORT, but you can combine other parameters by separating them with commas. If you select neither SHORT nor LONG, the default is SHORT if \$STDLIST is less than 99 characters and LONG otherwise.

**SHORT** Displays the file name, group name, account name, logical device number, sector address, reel number and file size (in sectors) of all files stored.

**LONG** Displays the same information as SHORT and the file record size, file type, end-of-file, file record limit, blocking factor, extents allocated, and maximum extents of all files stored.

**DATES** Displays the creation date, last access date, and last modification date of all files stored.

**SECURITY** Displays the creator and file access matrix of all files stored.

**OFFLINE** Sends file listings to the system line printer as well as your terminal. You can redirect the output from the system line printer to another output device with a FILE command redefining the formal file designator OFFLINE.

|                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>maxfiles</i>                  | Maximum number of files to store. If it encounters more than the maximum number of files, STORE prints an error message and quits. The default is 4,000.                                                                                                                                                                                                                                                                                                                                                                      |
| <i>accdate</i> or <i>moddate</i> | Stores only those files that have been accessed or that have not been modified since the date specified. The <i>accdate</i> parameter stores only files not accessed since the date specified. The <i>moddate</i> stores only those files that were modified on or after the date specified. The date is expressed in the numeric form <i>mm/dd/yy</i> . When no date is specified, files specified by the <i>filesets</i> are stored. If this option is included in an indirect file, it must precede all the fileset names. |
| ONERR[OR]                        | Specifies a tape error recovery mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| QUIT                             | STORE terminates when it encounters a tape error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| REDO                             | Instucts STORE to automatically recover from cartridge tape drive power failures and after all tape I/O errors.                                                                                                                                                                                                                                                                                                                                                                                                               |
| Default:                         | QUIT for labeled tapes. REDO for unlabeled tapes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| PURGE                            | Instructs STORE to purge files that are successfully stored.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| PROGRESS                         | Instructs STORE to peiodically display its progress as the percentage of the operation completed.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <i># minutes</i>                 | A positive integer representing the number of minutes. Specifies the interval at which STORE prints progress messages.                                                                                                                                                                                                                                                                                                                                                                                                        |
| STORESET                         | Describes multiple output devices. All devices must have the same device type, device subtype, density, and record size. You can only use STORESET if you purchase TurboSTORE.                                                                                                                                                                                                                                                                                                                                                |
| <i>tapelistspec</i>              | Describes output devices for the STORESET parameter. It has the following syntax:<br><br>$(tapelist) [ , (tapelist) ] [ \dots ] [ , tapelist ]$ where parentheses indicate devices grouped for parallel output.<br><br><i>tapelist</i> Designates a sequential group of output devices. It has the folowing syntax:<br><br>$tapedevice [ , tapedevice . . ] [ , tapedevice ]$ <i>tapedevice</i> Specifies an output device. Must backreference a FILE command.                                                                |

**Note**

When you specify only one *tapelist*, STORE numbers the reels you use beginning with 1. The first reel is numbered 1, the second is numbered 2, and so on. When you specify multiple *tapelists* (parallel backup devices), STORE numbers the reels from the first *tapelist* A1, A2, A3, and so on. It numbers the reels from the second *tapelist* B1, B2, B3, and so on.

## STORE

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INTER[leave] | Instructs STORE to store files in interleave format. File interleaving provides a higher sustained data rate from the disk. Use it to efficiently store files to multiple high-speed output devices. You can only use file interleaving if you purchase TurboSTORE.                                                                                                                                                                                                                                                        |
| COPYACD      | Specifies that access control definitions (ACDs) are to be copied. If you want the COPYACD option to store a file that has an ACD associated with it, the ACD is evaluated to determine if you have access to store the file and the ACD. (RACD permission is required to do this). If you do not have access to the ACD, the file is not stored. If you are granted access, the ACD is copied from disk to tape. If no ACD is associated with the file, the file is stored and an asterisk is printed after the filename. |

## Use

You can issue a STORE command from a session, job, or a program, but not in BREAK. It is breakable. You must have system manager (SM), system supervisor (OP), or privileged mode (PM) capability to store privileged files.

## Operation

The STORE command and its many parameters let you perform system backups (if you have system supervisor (OP) capability), back up individual files for your own records, and permanently store unused files to tape.

Users with system manager (SM) or system supervisor (OP) capability can store any file in the system. Users with account manager (AM) capability can store any file in their account (except files with negative file codes, which they can store only if they also have privileged mode (PM) capability).

The STORE command speeds system backups by copying files sequentially or simultaneously to a group of similar devices. Storing files to multiple devices significantly decreases the amount of time required to store a large set of files. STORE also provides a special format (called interleave format) for quickly storing files to fast backup devices. For more information on these STORE options, refer to *Storing and Restoring Files* (32033-90133).

Before entering a STORE command, identify the devices to which you intend to copy files with a FILE command. Backreference the devices in the STORE command.

You can issue the STORE command with the RUN command. Specify options, filesets, and keywords with the INFO=" " parameter. If you do not specify any INFO=" " parameters, MPE V/E displays an arrow (<-- ) prompt. In response, you can enter a complete STORE command, a complete RESTORE command, an MPE V/E command (preceded by a colon), or EXIT.

If you press **BREAK** while storing files, the operation continues while you issue other MPE V/E commands. You can use the ABORT command to cancel the store operation, or the RESUME command to resume receiving STORE messages.

## Examples

To copy all files from the group GP4X in your logon account to a tape, first assign the tape device a file name, then issue a STORE command. For example:

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

STORE sends a message to the system console requesting that the system operator mount the tape identified as "T". As it stores each file, it lists the file at your terminal. If a file is in use, STORE displays the name and the message "BUSY". Since you haven't specified an error recovery method, STORE uses the default. Below is a sample listing of the files stored:

| FILENAME.GROUP | .ACCOUNT  | LDN | ADDRESS  | REEL | SECTORS | CODE  |
|----------------|-----------|-----|----------|------|---------|-------|
| LABEL .MAC     | .TECHPUBS | 3%  | 01146363 | 1    | 257     | FIG   |
| MAIL .MAC      | .TECHPUBS | 1%  | 00120467 | 1    | 40      |       |
| MANPLAN4.MAC   | .TECHPUBS | 2%  | 01546471 | 1    | 844     | RASTR |
| MANPLAN5.MAC   | .TECHPUBS | 3%  | 01102561 | 1    | 544     | RASTR |
| FILES STORED:  | 4         |     |          |      |         |       |

To store all files that were modified on or after January 30th, 1989, and to instruct STORE to terminate if it encounters tape errors, enter:

```
:STORE @.@.*T;DATE>=1/30/89;ONERROR=QUIT
```

To store all files that are not in any PUB group and show the information on your terminal, enter:

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

To store file ABC and show all possible information on both the line printer and your terminal, enter:

```
:STORE ABC;*T;SHOW=LONG,DATES,SECURITY,OFFLINE
```

To store files using the specifications in the indirect file INDSTORE, use the following command:

```
:STORE !INDSTORE;*T
```

In this example, the indirect file INDSTORE contains the following:

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

To store files using the RUN command, enter:

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

```
<--STORE ABC
STORE/RESTORE, VERSION 3 (C) 1986 HEWLETT-PACKARD CO.
MON, JUN 10, 1989, 6:27 AM
```

```
FILES STORED: 1
FILES NOT STORED: 0
```

```
<--
```

## STORE

To store files to two sequential tape drives, first describe the tape drives in **FILE** commands. Backreference the **FILE** commands twice in your **STORE** command. For example:

```
:FILE TAPEA;DEV=TAPE;DEN=6250
:FILE TAPEB;DEV=TAPE;DEN=6250
:STORE @.PUB.TECHPUBS;SHOW;STORESET=(*TAPEA,*TAPEB)
```

To store files to three similar devices in parallel, use the following form of the **STORESET** parameter:

```
:FILE TAPEA;DEV=TAPE
:FILE TAPEB;DEV=TAPE
:FILE TAPEC;DEV=TAPE
:STORE @.PUB.TECHPUBS;STORESET=(*TAPEA),(*TAPEB),(*TAPEC)
```

To store files to two parallel groups of two sequential devices , use the following form of **STORESET**:

```
:FILE TAPEA1;DEV=TAPE
:FILE TAPEA2;DEV=TAPE
:FILE TAPEB1;DEV=TAPE
:FILE TAPEB2;DEV=TAPE
:STORE @.PUB.TECHPUBS;STORESET=(*TA1,*TA2),(*TB1,*TB2)
```

Add the **INTER** parameter to your **STORE** command to copy files in interleave format. For example:

```
:FILE TAPEA;DEV=TAPE
:FILE TAPEB;DEV=TAPE
:STORE @.PUB.TECHPUBS;SHOW;STORESET=(*TAPEA,*TAPEB);INTER
```

## Additional Discussion

*MPE V Storing and Restoring Files (32033-90133)*

*MPE System Backup and Recovery (32033-90134)*

## STREAM

Spools batch jobs or data from a session or job. The optional time-related parameters of the **STREAM** command may be used to schedule jobs.

### Syntax

$$\text{STREAM filename [ ,char ] [ ;AT=timespec ] \left[ ;\text{DAY}=\begin{cases} \text{day-of-week} \\ \text{day-of-month} \\ \text{days-until-month} \end{cases} \right] \\ [ ;\text{DATE}=mm/dd/yy ] [ ;\text{IN}= [ \text{days} ] [ ,\text{hours} [ ,\text{minutes} ] ] ]$$

### Parameters

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>filename</i>     | The <b>EDITOR</b> file containing the commands of the job.<br><br>The first character of the first record is assumed to be the replacement character for the expected colon (:) that identifies MPE V/E commands. You must have <b>EXECUTE</b> or <b>READ</b> access.                                                                                                                                                                                       |
| <i>char</i>         | Character used in place of the colon (:) to identify MPE V/E commands within the input file. When the input file is entered on a device configured to accept jobs, sessions, or data input via the <b>DATA</b> command, this character can be any ASCII special (non-alphanumeric) character except a colon. The default is an exclamation point (!).                                                                                                       |
| <b>AT</b>           | Absolute time specification (incompatible with <b>;IN=</b> ).                                                                                                                                                                                                                                                                                                                                                                                               |
| <i>timespec</i>     | Time specification. This is the absolute time of day in the format <b>HH:MM</b> where <b>HH</b> is the hour of the day ( $0 \leq \text{HH} \leq 23$ ) and <b>MM</b> is the minute of the hour ( $0 \leq \text{MM} \leq 59$ ). If <b>DAY</b> or <b>DATE</b> are not specified, <i>timespec</i> is the time of the current day. If <i>timespec</i> is earlier than the current time, the next day is implied (unless <b>DAY</b> or <b>DATE</b> is specified). |
| <b>DAY</b>          | Absolute day specification.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <i>day-of-week</i>  | Day of the week. Allowable values are:<br><br><div style="margin-left: 40px;"> <b>SUN</b>[<b>DAY</b>]<br/> <b>MON</b>[<b>DAY</b>]<br/> <b>TUE</b>[<b>SDAY</b>]<br/> <b>WED</b>[<b>NESDAY</b>]<br/> <b>THU</b>[<b>RSDAY</b>]<br/> <b>FRI</b>[<b>DAY</b>]<br/> <b>SAT</b>[<b>URDAY</b>] </div>                                                                                                                                                                |
| <i>day-of-month</i> | Day of the month. The integers 1 through 31. It indicates the calendar day of the month. If <i>dom</i> is greater than or equal to the current day-of-month, the current month is indicated. If <i>dom</i> is less than the current day-of-month, the next month is indicated. An error message is generated if the day-of-month does not correspond                                                                                                        |



## STREAM

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            | to the month (e.g. 31 is entered for February). If <i>dom</i> is omitted, the current date is used.                                                                                                                                                                                                                                                                                                                                                                                   |
| <i>days-until-monthend</i> | The negative integers -31 through -1. It indicates the calendar day on which the job will run, calculated from the end of the specified month. For example, a -1 value represents the last day of the month. If the specified day indicates a day earlier than the current day, the following month is assumed. For example, if today is the seventh day from the end of the month and a -8 value is entered, the job is scheduled for the eighth day from the end of the next month. |
| DATE                       | Absolute date specification.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <i>cusdate</i>             | Date, specified in the format <b>MM/DD/YY</b> , where <b>MM</b> is the month ( $1 \leq \text{MM} \leq 12$ ), <b>DD</b> is the day ( $1 \leq \text{DD} \leq 31$ ), and <b>YY</b> is the year. If omitted, the current date is used.                                                                                                                                                                                                                                                    |
| IN                         | Relative date or time specification.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <i>days</i>                | Days. A positive integer indicating the offset in days from the current date.                                                                                                                                                                                                                                                                                                                                                                                                         |
| <i>hours</i>               | Hours. A positive integer ( $0 < \text{hours} \leq 23$ ) indicating the offset in hours from the current time. If omitted, zero is used.                                                                                                                                                                                                                                                                                                                                              |
| <i>minutes</i>             | Minutes. A positive integer ( $0 \leq \text{minutes} \leq 59$ ) indicating the offset in minutes from the current time. If omitted, zero is used.                                                                                                                                                                                                                                                                                                                                     |

## Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** aborts execution and any partially streamed job.

## Operation

With the **STREAM** command, you can initiate jobs while in an interactive session by constructing your job from your terminal, or by reading records from a card, disk, or tape file. When the job is read, MPE V/E spools it onto a disk file, assigns it a job number, and processes it independently as an entity completely separate from your session. In the meantime, MPE V/E allows you to continue with your session.

You can initiate jobs in this way only if the system operator, or a user who has been given system operator capabilities, has enabled the MPE V/E stream facility by entering the console command **STREAMS**. The console command **STREAMS** also specifies a streaming device, which to MPE V/E appears to be the source of your job input regardless of the device you actually use for this input. As a result, the listing device that corresponds to the streaming device (and not your terminal) displays the job number assigned by MPE V/E and the listing generated by the job. To stream data to disk the input device used must be configured to accept the **DATA** command.

When you enter **STREAM** without an input file (that is, with the terminal as the default input device), during a session or a job, MPE V/E prompts you for input by displaying a greater than (>) character. When you enter **STREAM** for a device other than your terminal, MPE V/E does not print the prompt character.

## How to Stream Jobs

Begin each job in the input file with the **!JOB** command and terminate it with the **!E0J** command. Begin each data file with the **!DATA** command and terminate this file with the **!E0D** command. Begin all commands with an appropriate substitute character (other than a colon), as in **!JOB** or **#DATA**. When the input file is spooled to disk, MPE V/E replaces the substitute command identifier with a colon, so that the data files are properly interpreted when executed.

After reading the **!E0J** command that terminates the job, MPE V/E assigns each job a unique job number (**JobID**). MPE V/E also assigns each job a preset priority, unless you specify otherwise in the **JOB** command, and processes the job independently of the initiating job. Regardless of which device you use to submit the input file, all jobs in that file are treated as though they originated on the unique streaming device designated by the system operator (with the **STREAMS** command). The listing for each spooled job and the job number are written to the standard list device that corresponds to the streaming device. You may, however, use the **OUTCLASS=** parameter of the **JOB** command to direct the listing to another device.

## How To Time-Schedule Jobs

You may specify the time a job is to enter the **WAIT** state in absolute or relative time.

**Absolute**            You supply an exact time for the job using the **AT** parameter with or without the **DAY** or **DATE** parameter.

**Relative**            You specify a time offset from the current time using the **IN** parameter.

If the time specified is the same as the current time, the specified job logs on immediately. If the time specified is earlier than the current time, a warning message is generated. Otherwise, any time in the current century can be specified.

If no errors are detected, a **JobID** is displayed on the user's screen. If more than one job is included in the *inputfile*, each job is assigned a unique **JobID** and all of the jobs are scheduled at the same time. If the *inputfile* also contains **DATA** files, the files are immediately added to the input device directory (**IDD**). There is a limit of 64 jobs and **DATA** files in any one *inputfile*.

When a job is scheduled for a future time, it enters the **SCHED** state. When the specified time is reached, the job enters the **WAIT** state and is executed when system variables allow.

## Terminating Streamed Jobs

To terminate interactive job input, enter a colon (:). In response, MPE V/E ceases prompting for batch job input and instead prompts you for another command:

```
>: ** Denotes end of batch job input. **
:_ ** MPE V/E prompts for next command. **
```

Pressing **BREAK** aborts the **STREAM** command and any job currently being entered through the command. Incompletely spooled disk space is returned to the system.

If you make an error while entering the **JOB** command, you receive an error message on your job listing device. The system operator, however, receives no indication of the job or the error.

## STREAM

### Terminating Time-Scheduled Jobs

Jobs that have been scheduled for **STREAM** execution can be terminated with the **ABORTJOB** command. Refer to the *MPE V/E System Operation and Resource Management Reference Manual* (32033-90005), for information on using the **ABORTJOB** command to terminate time scheduled jobs.

### Special Considerations

In order to **STREAM** a file, you must have **READ** and **LOCK** access or **EXECUTE** access to that file. However, **READ** and **LOCK** access would allow general users to obtain security information within the file, such as passwords and lockwords. To allow general users to **STREAM** the file without giving them access to secure information, you may allow **EXECUTE** access only. For more information on file security, refer the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

#### Note



Scheduled jobs will survive a **WARMSTART**. Any other system starts cause scheduled jobs to be deleted. If the system is brought down for any reason, execute a **SHOWJOB** command to show the scheduled jobs. Then reschedule the jobs when the system is brought back up.

A scheduled job uses an entry in the **JMAT** table. Because of the limited recoverability of scheduled jobs, it is recommended that jobs be scheduled no more than a few days in advance.

If a user specifies a day or date for a job, but does not specify a time, the job does not enter the **WAIT** state at midnight on the specified day. Instead, it uses the time that the **STREAM** is executed, and enters the **WAIT** state at that time on the specified day.

### Examples

To stream a job from a disk file, you must name the input file in the **STREAM** command:

```
:STREAM ABC
```

If you use a character other than an exclamation point (!) as the substitute command identifier in your job input, you must identify that character in the **STREAM** command. Because you enter this character as the second positional parameter in this command, you must always precede it with a delimiting comma, even when you omit the input filename (the first parameter):

```
:STREAM,* ** Asterisk used as a substitute**
>*JOB MAC.TECHPUBS ** command identifier. **
>*FORTGO MYPROG
>*EOJ
#J74
>:
:_
:
```

## STREAM

If your job input file contains subsystem commands, such as commands directed to the editor, do not enter any command identifier *character* at the beginning of these commands. For instance, when using the Editor, enter the subsystem commands as follows:

```
:STREAM EXAMPLE ** The job input file is EXAMPLE. **
!JOB WXYZ,WRITER.TECHPUBS,MAC ** Initiates the job WXYZ. **
!EDITOR ** Invokes the Editor. **
TEXT ABC ** Subsystem commands without (!) as an identifier. **
LIST ALL,OFFLINE
EXIT
!EOJ ** Terminates job. **
#J87 ** Job number supplied by MPE V/E. **
:
```

If you want the job listing to appear on a device other than the standard listing device associated with the streaming device, you can specify this other device in the JOB command. Enter:

```
:STREAM
>!JOB MAC.TECHPUBS;OUTCLASS=12
```

The following section contains additional examples of STREAM command usage. For these examples, assume that the current date and time are Monday, April 17, 1989, 12:00 p.m. Also assume *jobfile* contains a valid STREAM job.

|                                            |                                                                                                                                                                                    |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <u>:STREAM JOBFILE</u>                     | JOBFILE is introduced immediately.                                                                                                                                                 |
| <u>:STREAM JOBFILE;AT=8:00</u>             | JOBFILE is introduced at 8:00 a.m., Tuesday, April 18.                                                                                                                             |
| <u>:STREAM JOBFILE;AT=20:00</u>            | JOBFILE is introduced at 8:00 p.m., Monday, April 17.                                                                                                                              |
| <u>:STREAM JOBFILE;IN=,8</u>               | JOBFILE is introduced in eight hours, at 8:00 p.m., Monday, April 17.                                                                                                              |
| <u>:STREAM JOBFILE;IN=1,8</u>              | JOBFILE is introduced in one day plus eight hours, at 8:00 p.m., Tuesday, April 18.                                                                                                |
| <u>:STREAM JOBFILE;DAY=MON;AT=8:00</u>     | Since the time specified (8:00 a.m.) is earlier than the current time, JOBFILE is introduced at 8:00 a.m., Monday, April 24.                                                       |
| <u>:STREAM JOBFILE;DAY=MONDAY;AT=20:00</u> | Since the time specified (8:00 p.m.) is later than the current time, JOBFILE is introduced at 8:00 p.m., Monday, April 17.                                                         |
| <u>:STREAM JOBFILE;DAY=18;AT=20:00</u>     | Since the day of the month (18) is later than the current day of the month (17), the current month is assumed. JOBFILE is introduced on Tuesday, April 18, at 8:00 p.m.            |
| <u>:STREAM JOBFILE;DAY=5</u>               | Since the day of the month (5) is earlier than the current day (16), the next month is assumed. Since no time was specified, JOBFILE is introduced on Friday, May 5, at 12:00 p.m. |

## STREAM

:STREAM JOBFILE;DAY=31

Since there is no April 31, the next month is assumed. Since there is a May 31, this is a legal command. **JOBFILE** is introduced on Wednesday, May 31, at 12:00 p.m. If there was no May 31, this would result in an error.

:STREAM JOBFILE;DAY-2

The "-2" means the second to last day of the month, and since no time was specified, the current time is used. **JOBFILE** is introduced on Saturday, April 29, at 12:00 p.m.

:STREAM JOBFILE;DAY-20

The "-20" means the twentieth day from the end of the month. If one assumes the current month, that implies April 11, but April 11 is earlier than the current day; therefore, the next month is assumed. **JOBFILE** is introduced on Friday, May 12, at 12:00 p.m.

:STREAM JOBFILE;DATE=4/16/84;AT=8:00

Since the specified time is earlier than the current time, this command is not legal and results in an error.

:STREAM JOBFILE;DATE=4/16/84;AT=20:00

The specified time is later than the current time, so this command is legal. **JOBFILE** is introduced on Monday, April 17, at 8:00 p.m.

## Additional Discussion

*MPE V System Operation and Resource Management Reference Manual (32033-90005)*

## STREAMS

Enables or disables the streams device: it permits users to submit job and/or data streams, or prevents them from doing so.

### Syntax

STREAMS { *ldev* }  
          { OFF }



### Parameters

- ldev*           The logical device number of the **STREAMS** device. This device must also have an output device number or class that references logical devices of type 32. Any input device except the system console or terminals may be used, providing that it was configured as job-accepting in the SYSDUMP dialogue.
- OFF            Disables the **STREAMS** facility.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You may issue it only from the console unless it is distributed to users with the **ALLOW** command.

### Operation

The system operator executes the **STREAMS** command after a startup to enable the **STREAMS** facility. The **STREAMS** device must be enabled each time the system is brought back online in order to allow users to stream jobs.

Streamed jobs are processed separately by MPE V/E, allowing users to continue with other work at their terminal. If the streamed job is submitted on a tape drive rather than from a terminal, MPE V/E processes it without requiring the user's attention. Any attempt to stream a job when the **STREAMS** facility is disabled generates the following message:

STREAM FACILITY NOT ENABLED: SEE OPERATOR. (CIERR 82)

The device normally configured as the **STREAMS** device is *ldev* 10. However, *ldev* 10 may not correspond to an actual device, such as a tape drive or card reader, physically connected to the computer. If this is the case, then the **STREAMS** device is considered a "pseudo-device". Regardless of whether the device physically exists or not, it must be entered into the I/O configuration table as a legitimate logical device. It must be assigned the device class **JOBTAPE**.

## **STREAMS**

### **Examples**

To enable jobs and data streams on logical device #10, enter:

```
:STREAMS 10
```

To disable data streams, enter:

```
:STREAMS OFF
```

### **Additional Discussion**

Refer to the discussion of the **JOB** and **STREAM** commands in this manual.

## SUSPENDSPOOL

Suspends output to a spooled device.

### Syntax

```
SUSPENDSPOOL ldev [;FINISH]
```

### Parameters

*ldev*                The logical device number of a spooled device.

FINISH             Directs the device to complete the currently active spool file and then stop.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You may issue it only from the console unless it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

### Operation

When the spooler process is suspended, the message "SP# *ldev* SPOOLER SUSPENDED" is displayed on the console. You may also determine the spooler's status by entering "SHOWOUT SP;JOB=@". If suspended, any spool files listed are ready for printing; none are active, and a SHOWDEV of the spooled device will indicate that the device is still spooled. Refer to the SHOWOUT command in this manual.

When suspending an active spool file, you should first take the output device offline. This gives you time to enter the command and determine that the active file is the one being printed. If you issue SUSPENDSPOOL without taking the device offline, that file might finish and another file start printing while you enter the command.

When your instruction has been sent to the spooler process, MPE V/E returns a colon prompt (:). The command is not executed, however, until the output device is returned online. Only then do you receive the "SPOOLER SUSPENDED" message.

### Examples

To suspend printing on logical device #6 (the line printer), enter:

```
:SUSPENDSPOOL 6
```

To suspend printing on logical device #6 once the currently active spool file is completely printed, enter:

```
:SUSPENDSPOOL 6;FINISH
```



---

## SWITCHLOG

Closes the current system log file, then creates and opens a new one.

### Syntax

SWITCHLOG

### Parameters

None.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You either require system supervisor (OP) capability to use it, or you must issue it from the console, unless it is distributed to users with the **ALLOW** command.

### Operation

When the SWITCHLOG command is executed, MPE V/E displays the previous system log file number (*xxxx*), the percentage of file space used (*yy*%), and the current open log file (*zzzz*), as shown in the following example:

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

If logging is enabled but currently suspended, MPE V/E displays the following message:

```
NO LOGGING
LOG FILE xxxx IS yy% FULL
```

---

### Note



Do not build new log files, since MPE V/E creates them automatically. If you do use **BUILD** to create a new log file, and then attempt to switch the current log file to the file you created, user logging will suspend in an error state. The following message is displayed:

```
LOG FILE NUMBER xxx ERROR #100.
LOGGING SUSPENDED.
```

---

### Example

To switch logging to a new log file, enter:

```
:SWITCHLOG
```

### Additional Discussion

Refer to RESUMELOG in this chapter, and to the section, “MPE Logging Facilities”, in the *MPE System Operation and Resource Management Reference Manual* (32033-90005).

## SYSDUMP

Starts the configurator dialog, and copies all system and user files to magnetic tape or serial disk and creates a backup of your system, user files, and configuration data.

### Syntax

```
SYSDUMP [$NULL
 dumpfile] [, auxlistfile]
```

### Parameters

|                    |                                                                                                                                                                                                                                                                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>\$NULL</b>      | Allows the system operator to step through the configuration dialog and alter system parameters without actually changing the system.                                                                                                                                                                                                               |
| <i>dumpfile</i>    | The destination file of the modified or duplicate system, previously defined with the FILE command. The <i>dumpfile</i> must backreference a magnetic tape or serial disk file.                                                                                                                                                                     |
| <i>auxlistfile</i> | Actual file designator of the output device file to which all listings requested during the configurator/user dialog are written. The formal file designator used by the Segmenter for this file (when it is invoked to add or replace SL segments) is SEGLIST. The default is \$STDLIST. It may backreference a magnetic tape or serial disk file. |

If you omit all the parameters and use the SYSDUMP command alone, the implied file is assumed to be *dumptape*. The device will be TAPE, if it exists on the system, or CTAPE. If neither exists, you get an error.

### Use

You may issue this command from a session or a job, but not in BREAK or from a program. Pressing **BREAK** suspends execution. It requires system supervisor (OP) capability.

### Operation

When the SYSDUMP command is executed, you must direct MPE V/E to write the output to a specific device. To do so, precede the SYSDUMP command line with two file equations. Then backreference these files by preceding the *dumpfile* parameter and the *auxlistfile* parameter with an asterisk (\*) on the SYSDUMP command line. To associate *dumpfile* with a tape drive, for example, enter:

```
:FILE DUMP;DEV=TAPE
```

Similarly, to produce the output listing from the SYSDUMP command on a line printer, enter:

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

In general, use the following format for the FILE command:

```
FILE formaldesignator [= filreference] ; DEV = device [; REC = recsize [DEN = density]]
```

where:

- *device* specifies the device class name or *ldev* number of a tape drive or serial disk.

## SYSDUMP

- *recsize* determines the size of the blocks on the store tape. The record size may be any multiple of 256 words, with a maximum record size of 8192 words for tape or serial disks.
- *density* specifies the tape density in characters-per-inch (CPI). This parameter is only used for variable-density tape drives. The values allowed for tape drives are: HP 7976A - 1600, 6250; HP 7974 - 800, 1600; HP 7978 - 1600, 6250.

Once you have designated the peripheral devices that *dumpfile* and *auxlistfile* refer to, issue the SYSDUMP command, preceding each filename with an asterisk (\*) to indicate a backreference to the definitions in the FILE commands.

The SYSDUMP command invokes the STORE command after writing system information to the beginning of the tape. Progress messages appear once the system begins writing user files to the tape.

The default interval is one minute; however, during SYSDUMP dialog, you may specify your own preferred interval, in minutes, by using the PROGRESS keyword in response to the "ENTER DUMP FILE SUBSETS" prompt. Refer to the STORE command in this chapter.

---

### Note



To start the configurator dialog without copying data to tape, use the system-defined file \$NULL as the *dumpfile*. MPE V/E accepts this file specification and opens the file as though it were a standard file, but performs no corresponding WRITE operation. SYSDUMP \$NULL allows you to examine the current configuration, and even make changes to it, without affecting the system in any way. It terminates with an error message, which you may ignore.

---

### Example

Enter SYSDUMP to begin dialog with the MPE V/E configurator. Once the dialog is complete, configuration changes are copied to the DUMP file, and any listings requested during the dialog are output to the file named LIST. In the example below, the first file equation directs the output of the SYSDUMP command to a tape drive. The second file equation sends any listings generated during the configuration dialog to a line printer.

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

During SYSDUMP dialog you must specify an interval, in minutes, for progress messages (the default is one minute):

```
:SYSDUMP *T
ANY CHANGES? NO
ENTER DUMP DATE? 8/15/85
ENTER DUMP FILE SUBSET(S) 0.0.0;FILES=20000;PROGRESS=3
```

### Additional Discussion

For a full discussion of the SYSDUMP command, refer to the sections entitled "The System Supervisor" and "Configuring your System" in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

---

## TAKE

Releases a device previously assigned to diagnostics so that it may be brought online.

### Syntax

```
TAKE ldev
```

### Parameter

*ldev*                    The logical device number of the device taken from diagnostics.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. You may issue it only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

### Operation

After diagnostic testing, you must issue the **TAKE** command to release the device from diagnostic control. Only then can it be brought online with the **UP** command.

### Example

To take logical device #35 from diagnostics and bring it back online, enter:

```
:TAKE 35
:UP 35
:SHOWDEV 35
```

The screen then shows the following display:

| LDEV | AVAIL   | OWNERSHIP | VOLID | ASSOCIATION |
|------|---------|-----------|-------|-------------|
| 35   | A AVAIL | UP        |       |             |

---

## TELL

Sends a message to another session.

### Syntax

$$\text{TELL} \left\{ \begin{array}{l} [ \# ] Snnn \\ [ sessionname, ] username.acctname \\ @ \\ @.acctname \\ @S \end{array} \right\} [ text ]$$

### Parameters

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [#]Snnn              | The session number as assigned by MPE V/E. This session number receives the TELL message.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| username.accountname | The name of the session or user who should receive the message, and the account name to which the message is directed. This parameter is the same as the session identity entered with the HELLO command. You can list all the <i>user.accountnames</i> to which you may direct a TELL message by issuing a SHOWJOB command. Users who have issued a SETMSG OFF command are listed as being in QUIET mode and will not receive your TELL message. If several users are running under the same session identity, MPE V/E sends the message to all of them. |
| @                    | All sessions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| @.acctname           | All sessions under the account name established by the system manager.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| @S                   | All sessions. This is the same as the @ parameter.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| text                 | Message text, preceded by a space and consisting of any string of ASCII characters. The default is that no text is printed; however, MPE V/E still prints the FROM message as follows:                                                                                                                                                                                                                                                                                                                                                                    |

FROM/sessionid

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command.

## Operation

The TELL command transmits a message from the sender's job or session to one or more sessions currently running. The message appears on the receiving session's list device. Messages sent via this command may include escape and control characters that invoke bells or inverse video. If you send a message to a terminal that is currently interacting with a program, MPE V/E queues the message as high as possible among the current input/output requests but does not interrupt reading or writing in progress. If the session or user designated to receive the message is not running, or if the job is spooled, the transmitting job/session receives a system message indicating this. MPE V/E blocks the TELL command if the receiving session (including the system console) is operating in QUIET mode (refer to SETMSG command) and informs the sender via the following message:

*Snnn username.acctname* NOT ACCEPTING MESSAGES

### Note



TELL messages can be sent to a session only. If you send a TELL message to the system console while it is running in QUIET mode, it will not be received. To override QUIET mode, use the TELLOP command, which delivers messages to devices, including the device designated as the system console.

You cannot send TELL messages to a job or to yourself, only to a session. If you try to send a message to a job the following warning is issued:

#Jnn NOT INTERACTIVE; MESSAGE NOT SENT TO JOB (CIWARN 1627)



## Examples

To send a message to a user identified as BROWN who is logged on under account A and running a session named BROWNSES, telling Brown to use a particular file, enter:

:TELL BROWNSES,BROWN.A USE FILEX

To send a message asking all users logged on in account A to log off, enter:

:TELL @.A PLEASE LOG OFF

---

## TELLOP

Sends a message to the system console.

### Syntax

TELLOP [*text*]

### Parameter

*text* Message text, preceded by a space and consisting of any string of ASCII characters. The default is that no text is printed; however, MPE V/E still prints the **FROM** message as follows:

**FROM**/*sessionid*

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command.

### Operation

The TELLOP command sends a message to the system console. The message text appears on the system console, preceded by the time it was transmitted and your job or session number. Like messages transmitted between users (see the **TELL** command), this message is printed as soon as possible without interrupting any console input/output currently in progress. The message is sent to the device designated as the system console even if no session is logged on, or if an active session is running in **QUIET** mode.

### Example

To send a message to the system operator with instructions to mount a tape, enter:

:TELLOP PLS MOUNT MYTAPE,VERSION 1

## TUNE

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

### Syntax

$$\text{TUNE [ minclockcycle ]; } \left\{ \begin{array}{l} \text{CQ} \\ \text{DQ} \\ \text{EQ} \end{array} \right\} = [ \text{base[ , limit[ , min[ , max[ ; ... ] ] ] ] ]$$

### Caution



Misuse of this command can significantly degrade the operating efficiency of the system.

### Parameters

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>minclockcycle</i> | Minimum number of milliseconds allotted for the replacement algorithm to cycle through memory. If it cycles through memory in less time, memory allocation is delayed and prevents disk thrashing.                                                                                                                                                                                                                                                                                                                                                                                        |
| <i>base</i>          | Priority at which processes that execute in the CS, DS, or ES subqueues begin in their respective queues. MPE V/E uses these processing queues, also referred to as “circular subqueues”, to differentiate processes according to the amount of CPU processing time allotted to each. A process executing in the ES subqueue, for example, will be initiated only if no higher-priority processes are ready to run. As a result, a process in the ES subqueue may execute more slowly. In general, the more critical the process, the higher the subqueue to which it should be assigned. |
| <i>limit</i>         | Highest priority that processes executing in the CS, DS, or ES subqueues can attain.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <i>min</i>           | The minimum number of milliseconds (filter value) for which a process may use the CPU before its priority is reduced.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <i>max</i>           | The maximum number of milliseconds (filter value) for which a process may use the CPU before its priority is reduced. The value of <i>max</i> must be greater than or equal to the value of <i>min</i> .                                                                                                                                                                                                                                                                                                                                                                                  |

### Use

You may issue this command from a session, job, or in **BREAK**, but not from a program. Pressing **(BREAK)** suspends execution. Its use requires system supervisor (OP) or system manager (SM) capability.



## TUNE

### Operation

The system supervisor uses the **TUNE** command to change the filter or priority limits of a circular subqueue and/or the value of the manager algorithm threshold. The command is used primarily in online tuning, so that the system can manage the current processing load efficiently.

For processes executing in the CS subqueue, the parameters *min* and *max* refer to the absolute limits of the “average short transaction time”, or filter. This filter is recomputed after every transaction is completed, and then compared against the CPU time of each process to determine whether the priority of the process should be decreased. By contrast, processes executing in the DS or ES subqueues use a fixed value rather than an average. The same value is used for both queues and is specified by setting *min* and *max* equal to the desired value. Both *min* and *max* must be specified, but if they are not given the same value, the value of *max* is used.

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

When a process begins executing in the CS subqueue, it is given a priority of CBASE. When a process stops (for disk I/O, terminal I/O, or preemption), its new priority is determined so that it may be requeued for the CPU. If the process has completed a transaction, defined as the time between terminal reads, its priority becomes CBASE and the value of an “average short transaction” is recalculated. If the CS process has not completed a transaction, and if the process has exceeded the filter value since its priority was last reduced, the priority is decreased (but will not exceed the CLIMIT).

DS and ES processes begin at DBASE and EBASE respectively, and are rescheduled according to the same criteria used for the CS process. However, a fixed value (the value of *max*, specified for the subqueue) is used in place of the “average short transaction time”, which is used for CS processes only. If the values specified for *max* in the CS, DS, and ES queues are too large, system response may become erratic. If they are too small, excessive memory management may occur, because process-swapping occurs too frequently. In either case, system performance is degraded. The recommended settings for *min* and *max* for the CS subqueue are 0 and 300 respectively, since short transactions are favored in most cases. A value for *min* and *max* of 1000 usually produces efficient system operation for the DS and ES subqueues.

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

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

The following defaults are set when the system is cold-loaded from the system disk (a WARMSTART or COOLSTART).

*minclockcycle:* 1000

CQ *base:* 152  
*limit:* 200  
*min:* 0  
*max:* 300

DQ *base:* 202  
*limit:* 238  
*min:* 1000  
*max:* 1000

EQ *base:* 240  
*limit:* 253  
*min:* 1000  
*max:* 1000

### Example

To set the minimum clock cycle to 1000, CQ *base* to 152, *limit* to 200, and CQ maximum filter to 300; and DQ *base* to 202, DQ *limit* to 238, DQ (and EQ ) minimum and maximum filter to 1000, enter:

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

### Additional Discussion

For additional information about process scheduling, refer to the discussion on “Process Scheduling” in chapter 3 of the *MPE V Intrinsic Reference Manual* (32033-90007).

---

## UP

Returns a particular device to its normal function on the system by cancelling a **DOWN** command that had been issued for the device.

### Syntax

**UP** *ldev*

### Parameters

*ldev*                The logical device number of the device being returned to service online.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **(BREAK)** has no effect on this command. It may be issued only from the console *unless* it is distributed to users with the **ALLOW** or **ASSOCIATE** commands.

### Operation

The **UP** command makes available to users a device previously taken offline with the **DOWN** command. Ownership of the device is not affected by this command. If a device is owned by the system at the time it is downed, the system retains ownership even after the **UP** command is executed.

### Example

To allow logical device #10 to function again, enter:

```
:UP 10
:SHOWDEV 10
LDEV AVAIL OWNERSHIP VOLID ASSOCIATION

10 A AVAIL
```

### Additional Discussion

Refer to the **DOWN** command in this manual.

## VINIT

Runs the VINIT subsystem to format, initialize, and label private volumes, serial disks, and foreign disks. The prompt in the VINIT subsystem is ">".

### Syntax

```
VINIT [listdevice]
```

### Parameters

*listdevice*      Name of the device to receive output generated by the VINIT subsystem. This parameter must backreference a FILE command that names the device. The default is \$STDLIST.

### Functions

```
>COND ldev [;RECOVER]
```

Condenses free disk space and recovers lost disk space for private volumes.

```
>COPY fromldev, toldev [;GEN=[genindex]
 ;VER[IFY]
 ;BUF=size]
```

Copies the contents of one volume to another.

```
>DSTAT [ldev
 ALL
 @]
```

Displays the status of disk drives.

```
>DTRACK ldev
```

Processes suspect tracks in the defective track table and defective sectors in the defective sectors table. As you use DTRACK, you can either recover (that is, ignore), delete, or reassign the defective tracks.

```
>EXIT
```

Exits the VINIT subsystem.

```
>FOREIGN ldev
```

Makes a disk volume foreign by filling the label area of a volume with zeros.

```
>FORMAT ldev [;IBM[:volid]]
```

Allows online formatting of disk packs.

```
>HELP
```

Prints a listing of the VINIT commands, including syntax and descriptions, at \$STDLIST.

## VINIT

```
>INIT vname,ldev [,vsname,group,acct]
 ;GEN=genindex
```

Completes the conditioning of a private volume previously formatted by the SLEUTHSM program or the >FORMAT command.

```
>PDEFN [*.group.account vsname.group.account]
```

Lists the specified volume set. The asterisk (\*) specifies the home volume set. The default is the home volume set.

```
>PDTRACK ldev
```

Prints the contents from the defective tracks table or the defective sectors table at \$STDLIST.

```
>PFSPACE { ldev [;ADDR] }
 ALL
```

Prints the disk free-space map at \$STDLIST.

```
>PLABEL ldev
```

Prints the disk label at \$STDLIST.

```
>SCRATCH ldev [;RESET]
```

Writes (resets) a scratch-type label.

```
>SERIAL ldev
```

Writes a serial disk-type label.

```
>VERIFY ldev [;DTT]
```

Verifies (reads) data.

Each of the parameters referenced in the previous VINIT functions is defined below and discussed in detail under the heading "Private Volumes" in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

*ldev*            The logical device number of the disk drive on which the volume is mounted.

*fromldn*       The logical device number of the disk from which data is copied. Unless the device is a system domain disk, you must first remove it from normal system use by issuing a DOWN command for the device.

*toldn*          The logical device number of the disk volume to which data is copied. The device must be in the DOWN state. You must first remove the device from normal system use by issuing a DOWN command for the device.

*genindex*      A value from 0 to 32767 inclusive, indicating the generation index of the disk volume. If not specified, the *genindex* of *toldn* will be one greater than the *genindex* of *fromldn*. This parameter is valid only for private volumes.

VER[IFY]       Checks the readability of data on the destination drive. If an error is detected, VER[IFY] directs the VINIT subsystem to attempt to recover the track. If it is unrecoverable, the track is reassigned and VINIT will attempt to copy data once again. The VERIFY keyword may be abbreviated as VER.

|                       |                                                                                                                                                                                                    |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>size</i>           | The internal buffer size, which may be a value from 1 to 32 inclusive.                                                                                                                             |
| IBM[: <i>volid</i> ]  | Indicates that the disk is a single-sided IBM-formatted diskette. If <i>volid</i> is specified, it denotes an IBM volume identifier other than the default <i>volid</i> of IBMIRD.                 |
| <i>vname</i>          | The name of the disk volume that is a member of the volume set.                                                                                                                                    |
| <i>vsname.group</i> . | The name of the volume set.                                                                                                                                                                        |
| <i>acct</i>           |                                                                                                                                                                                                    |
| ADDR                  | Displays the disk free space entries for the specified logical device.                                                                                                                             |
| ALL or A              | Displays the disk free-space map in table format for all systems and private volumes that are physically mounted.                                                                                  |
| RESET                 | Resets a volume from scratch to non-scratch.                                                                                                                                                       |
| DTT                   | Directs VINIT to verify only suspect sectors and/or tracks as recorded in the defective sectors and/or tracks tables. This option is not available for serial or foreign disks or cartridge tapes. |

## Use

You may issue this command from a session or job, but not in **BREAK** or from a program. Pressing **BREAK** suspends execution. You require system supervisor (OP) or system manager (SM) capability to use this command.

## Operation

The VINIT command accesses the VINIT subsystem, which includes commands for online initialization, labeling, and formatting of private volumes.

## Example

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

---

**Note** In JOB mode, the VINIT subsystem can be invoked only as follows:



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

---

## Additional Discussion

Refer to VINIT in “Subsystems and Utilities” of the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).

---

## VMOUNT

Enables or disables MPE V/E's private volumes facility.

### Syntax

$$\text{VMOUNT } \left\{ \begin{array}{l} \text{ON } [ , \text{AUTO} ] \\ \text{OFF} \end{array} \right\} [ ; \text{ALL} ]$$

### Parameters

- ON or ON,AUTO** Enables the private volume facility so that all valid user **MOUNT** and operator **LMOUNT** requests are allowed. When **ON** is used without **AUTO**, the operator must reply to all **MOUNT** requests. When **ON,AUTO** is used, MPE V/E attempts to satisfy user **MOUNT** and operator **LMOUNT** requests without operator intervention.
- OFF** Requests to use the private volumes facility are rejected.
- ALL** Prints all private volumes console messages that are mount-related, including those not requiring operator intervention, on the console.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. It may be issued only from the console *unless* it is distributed to users with the **ALLOW** command.

### Operation

If the private volumes facility is enabled when you issue a **VMOUNT OFF** command, users with mounted volume sets are not affected; the command is satisfied when the last access is complete.

Once the private volumes facility has been enabled, use the **DSTAT** command to determine which volumes are actually mounted, and the **VSUSER** command to determine which users are using particular volume sets. Refer to the **DSTAT** and **VUSER** commands in this chapter.

The private volumes facility is disabled immediately following a system startup (the setting is equivalent to **VMOUNT OFF;ALL** ). However, you still receive console messages concerning private volume requests.

The operator has the greatest interactive control over the use of private volumes by using **VMOUNT ON;ALL**. The command that least interrupts the operator when users are accessing private volumes is **VMOUNT ON,AUTO**.

## Examples

To disable the private volumes facility so that no messages are sent to the console when users attempt to mount private volumes (the default condition), enter:

:VMOUNT OFF

To disable the private volumes facility and still receive messages on the console when users attempt to mount private volumes, enter:

:VMOUNT OFF;ALL

## Additional Discussion

Refer to the discussion on “Private Volumes” in the *MPE V System Operation and Resource Management Reference Manual* (32033-90005).



---

## VSUSER

Lists all users of a currently mounted, nonsystem volume set.

### Syntax

```
VSUSER [vsname]
```

### Parameter

*vsname*            A fully qualified volume set name. The default is that information for all currently mounted volume sets is listed.

### Use

You may issue this command from a session, job, or in **BREAK**, but not from a program. Pressing **(BREAK)** has no effect on this command.

### Operation

The **VUSER** command lists all users who have explicitly or implicitly mounted a non-system volume set. It also displays the volume set name, job number, and the group and account names of all users currently performing a **MOUNT** function. The **MOUNT** and **DISMOUNT** commands enable you to perform an explicit mounting; the **FOPEN/CLOSE** intrinsics enable you to perform an implicit mounting. You must have **UV** (use volumes) or **CV** (create volumes) capability to use this command. Refer to the **MOUNT** and **DISMOUNT** commands in this chapter.

### Example

To list all of the currently mounted volume sets, enter:

```
:VSUSER
```

| VOLUME SET NAME | JOBNUM | JOBNAME    |
|-----------------|--------|------------|
| -----           | -----  | -----      |
| USER.MANAGER    | #S260  | NORMA.MPEM |

### Additional Discussion

*MPE V System Operation and Resource Management Reference  
Manual (32033-90005)*

*MPE V Intrinsics Manual (32033-90007)*

## WARN

Sends an urgent message to jobs or sessions.

### Syntax

$$\text{WARN} \left\{ \begin{array}{l} @ \\ \left[ \begin{array}{l} \# \\ \# \end{array} \right] Jnn \\ \left[ \begin{array}{l} \# \\ \# \end{array} \right] Snnn \\ \left[ jsname, \right] user.acct \end{array} \right\} [ ; message ]$$

### Parameters

|                      |                                                                                                                                                                                                       |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @                    | All users receive the message, whether QUIET or non-quiet.                                                                                                                                            |
| #Jnn                 | A job number (assigned by MPE V/E) for the job that is to receive the message.                                                                                                                        |
| #Snn                 | A job number (assigned by MPE V/E) for the job that is to receive the message. Only jobs submitted on interactive devices can receive messages.                                                       |
| jsname,<br>user.acct | The names of the job or session and user who are to receive the message, and the account name under which they are running. (These names are the same as those entered via the JOB or HELLO command.) |
|                      | If several users are running under the same job or session identity, MPE V/E will send the message to all of them.                                                                                    |
| text                 | The message text, consisting of any string of ASCII characters no more than 67 characters. The message is terminated by a <b>Return</b> . The default is that no message is printed.                  |

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. It may be issued only from the console *unless* it is distributed to users with the **ALLOW** command.

### Operation

Sends an urgent message, interrupting any **READ** or **WRITE** in progress. The message will appear on the list devices of all sessions (even those that are **QUIET**) as:

OPERATOR WARNING: *text*

Messages sent with the **WARN** command will be received by jobs only if the job was submitted on an interactive device.

You have the option of running your session in **QUIET** mode. In that case, **TELL** messages from other users will be suppressed. **WARN** messages generated at the system console, however, override **QUIET** mode.

## WARN

---

### Note



Use caution when sending a warning to users. The **WARN** command overrides a block mode screen.

---

### Example

To send a **WARN** message to all sessions, followed by a **WARN** message to session #S51 , enter:

```
:WARN @;THE SYSTEM WILL SHUTDOWN IN 5 MINUTES. PLS LOG OFF.
:WARN #S51;LAST CHANCE TO LOG OFF GRACEFULLY.
```

---

## WELCOME

Defines the welcome message.

### Syntax

```
WELCOME [welcfile]
```

### Parameters

*welcfile*            An ASCII file containing the new welcome message.

### Use

You may issue this command from a session, job, in **BREAK**, or from a program. Pressing **BREAK** has no effect on this command. It may be issued only from the console *unless* it is distributed to users with the **ALLOW** command.

### Operation

The operator uses the **WELCOME** command to compose the message that is transmitted to users when they initiate jobs and sessions. The message is retained when you “warm-start” or “cool-start” the system, or choose the **UPDATE** restart option.

To define the welcome message, enter “**WELCOME**” and press “**Return**”. When the “#” prompt is displayed, begin entering the text of the message. You may use any number of lines, but the length of any line cannot exceed 72 characters. To terminate the message and complete the command, enter “**Return**” at the “#” prompt.

To define the welcome message from an editor file, you merely specify the *welcfile* parameter. Subsequent changes to the editor file do not affect the welcome message until you issue another **WELCOME** command with the specified filename.

---

**Note**            If no parameter is specified, you are prompted to enter the new message interactively.



To delete the old welcome message, issue the **WELCOME** command and enter **Return** at the “#” prompt.

### Example

To create a multi-line welcome message, enter:

```
:WELCOME
WELCOME TO THE HP3000 COMPUTER SYSTEM.
FILES WILL BE STORED EACH DAY BETWEEN 6AM AND 7AM.
RETURN
```





## **Section Divider**

### **3. User-Defined Commands**



## User-Defined Commands

---

MPE V/E allows you to define your own commands to fit specific needs. To build a user-defined command, or UDC, you may use **Editor** or another ASCII text editor to combine several standard MPE V/E commands into a single list. Assign a name to that list and save it. To invoke the UDC, enter its name at the MPE V/E colon prompt (:) and press **Return**: all the commands in the UDC list will execute. A user-defined command may be used in most instances where a standard MPE V/E command may be used, even within another UDC. User-defined commands cannot, however, be executed through the **COMMAND** intrinsic, or from within subsystems, unless you are in **BREAK**.

The UDC definitions are stored in one or more files. These files can exist at one of three different levels: user, account, or system. Any user may create UDCs for personal use. They may also be created by the account manager for anyone using the account, or by the system manager for users on the system. System-wide UDCs must be defined in a file that is released for all users to use.

When identically named UDCs occur at different levels, the user-level UDC takes precedence over the account-level UDC, which in turn takes precedence over the system-level UDC. For this reason, it is good practice to give your UDCs unique names. Exceptions to this rule, nested and logon UDCs, are discussed later in this chapter.

You must have special capabilities to establish account or system-level UDCs. For information on system manager, account manager, and system supervisor capabilities, refer to the *MPE V system Operation and Resource Management Reference Manual* (32002-90005).

### Note



No UDC name may consist of, or start with, the characters "RFA", since "RFA" is reserved for internal use. Names like "RFAST" are thus illegal. Although the **HELP** subsystem shows such UDCs in their correct form, the command interpreter does not recognize them and will issue the error message "UNKNOWN COMMAND NAME (CIERR 975)".

The **SHOWCATALOG** command lists the UDCs currently in effect, the level at which they are defined, and the file in which they reside; for example:

```
:SHOWCATALOG ** Name of file containing UDC definitions **
 ** UDC name and level **
UDCU.TOM.MPE
 WELCOME USER
 T USER
UDCA.PUB.MPE
 LIST ACCOUNT
 FCOPY ACCOUNT
UDCS.PUB.SYS
 R SYSTEM
```



If a UDC name that you do not recognize appears in the catalog, use the **HELP** command to discover the content of that UDC, unless **OPTION NOHELP** has been specified for that UDC (refer to “Execution Options” further on in this chapter). Suppose, for example, that **SHOWCATALOG** reveals an unfamiliar UDC named **SJ**. To find out exactly what that UDC contains, enter:

```
:HELP SJ **The UDC name used as a parameter of HELP **

USER-DEFINED COMMAND:

SJ
SHOWJOB
:
```

---

## Syntax of UDCs

A user-defined command has two parts: a header section containing control information, and a body consisting of MPE V/E commands.

### Header

The control information in the header section of a UDC consists of:

- The command name, composed of alphanumeric characters (16 maximum) and beginning with an alphabetic character. You cannot use the reserved word “RFA” as a UDC name or at the beginning of a UDC name.
- Parameters and their optional default values (maximum 16).
- Execution options (**LIST/NOLIST**, **BREAK/NOBREAK**, **LOGON/NOLOGON**, **HELP/NOHELP**).

### The Command Name

The first line (including continuation lines) in the header of a UDC gives the command interpreter the name of the command, any parameters, and the defaults for those parameters. A minimum first line could be:

A

In this example, the command is named **A**. It has no parameters. UDC headers may be more complex if you choose to specify options or parameters. A UDC header may extend over more than one physical line and may contain from 72 to 320 characters. Each line to be continued must end with an ampersand (&) as the last nonblank character.

### Parameters

UDC parameters are variables that are passed from the calling command (the UDC name) to the body of the UDC. This allows a general UDC to serve different specific purposes.

## Execution Options

There are several options that you may wish to specify in the UDC header. The options or their defaults are invoked when the UDC executes. The options available for use in a UDC declaration are explained in the following pages.

### LIST/NOLIST

The **LIST** option of a command allows the text of the UDC body to be listed on the standard list device after all parameters have been substituted. With the **LIST** option, errors are reported with a caret (^) under the error. The default is **NOLIST**. An example is:

```
:TDPOFFLINE
OPTION LIST
FILE LP;DEV=EPOC;CCTL;ENV=LP2.ENV2680A.SYS
RESUME
```

When the UDC is used, the text of the body is printed:

```
:TDPOFFLINE
FILE LP;DEV=EPOC;CCTL;ENV=LP2.ENV2680A.SYS
RESUME
READ pending
```

### BREAK/NOBREAK

If you specify **NOBREAK**, pressing **BREAK** has no effect on the command. If the **NOBREAK** option is not specified, all commands that can be broken remain subject to **BREAK**. The default is **BREAK**.

If a nonprogram command is broken, the UDC terminates. If a program command is broken and you use the **RESUME** command to resume the program, the UDC will also resume. An exception to this occurs if the **SETCATALOG** command is used while you are in the **BREAK** facility. In this case, the UDC is not resumed. Refer to “Using the **SETCATALOG** Command” in this chapter.

The UDC **NOBREAK** option overrides the **CAUSEBREAK** intrinsic. A program containing **CAUSEBREAK** does not break if it is executed from a UDC that specifies the **NOBREAK** option.

Suppose your UDC file contained the following:

```
S
SHOWJOB

SJ
OPTION NOBREAK
SHOWJOB

```

If you use the “S” UDC, you can break the listing as follows:

```
:S
JOBNUM STATE IPRI JIN JLIST INTRODUCED JOB NAME
#S55 EXEC 20 20 MON 2:49P FIELD.SUPPORT
#S57 EXEC ** BREAK pressed here. **
:
:RESUME
COMMAND ONLY ALLOWED IN BREAK.(CIWARN 1686) **BREAK aborts SHOWJOB.**
:
```

If you use the “SJ” UDC, you cannot break the listing. Enter:

```
:SJ

JOBNUM STATE IPRI JIN JLIST INTRODUCED JOB NAME
#S55 EXEC 20 20 MON 2:49P FIELD.SUPPORT
#S57 EXEC 52 52 MON 2:50P BOB.DATAMGT

2 JOBS:
 0 INTRO
 0 WAIT; INCL 0 DEFERRED ** BREAK pressed here has no effect **
 2 EXEC; INCL 2 SESSIONS
 0 SUSP
JOBFENCE= 2; JLIMIT= 2; SLIMIT= 16
```

### LOGON/NOLOGON

The LOGON option specifies that the UDC is automatically executed when you log on. There are three levels in the UDC hierarchy: user, account, and system. At most, one logon UDC at each level is executed. For users who have logon UDCs in each of the levels, the system logon UDC executes first, then the account logon UDC, and finally the user logon UDC. If more than one logon UDC exists within any level, only the first logon UDC will execute for that level without affecting the execution of logon UDCs at other levels. The default is NOLOGON. The first UDC at any level is the UDC named first in the SETCATALOG command for that level. The UDC file *ufile01* will be the first after the command SETCATALOG*ufile01*, *ufile02*, *ufile03* is executed.

For example, suppose your UDC was:

```
S
OPTION LOGON
SHOWJOB

```

When you logon, "S" automatically executes:

:HELLO BARBARA.USERS

HP3000 / MPE V G.03.00 (BASE G.03.00). MON, AUG 14, 1989, 8:43 AM

| JOBNUM | STATE | IPRI | JIN | JLIST | INTRODUCED | JOB NAME      |
|--------|-------|------|-----|-------|------------|---------------|
| #S55   | EXEC  |      | 20  | 20    | MON 6:43A  | FIELD.SUPPORT |
| #S57   | EXEC  |      | 52  | 52    | MON 6:45A  | BARBARA.USERS |

2 JOBS:

0 INTRO

0 WAIT; INCL 0 DEFERRED

2 EXEC; INCL 2 SESSIONS

0 SUSP

JOBFENCE= 6; JLIMIT= 7; SLIMIT= 40

## HELP/NOHELP

The HELP subsystem may be used to obtain information about user-defined commands, unless NOHELP is specified as an option. The default is HELP. The MPE V/E HELP subsystem does not contain a listing of UDCs. Therefore, if you enter the HELP subsystem first, you defeat the purpose of trying to list the contents of a UDC. Instead, enter *HELP udcname*. This forces the HELP subsystem to search out cataloged UDCs.

The HELP subsystem responds with a listing of the UDC:

:HELP GRAPHOUT

USER-DEFINED COMMAND:

GRAPHOUT

FILE GRAPHOUT;DEV=EPOC

RUN GRAPH.PUB.SYS

:

The following example illustrates the result of an attempt to use the HELP command on a UDC for which the NOHELP option has been specified:

:HELP S \*\* The "S" UDC contains a NOHELP option \*\*

Can't find anything under this command or in Table of Contents.

---

## Note



If you specify both the NOLIST and NOHELP options and an error occurs, the error is reported, but the line containing the error is not echoed.

---

## Body

As in the UDC header, any logical line in a UDC header may continue over more than one physical line and may contain as many as 320 characters. Each line to be continued must end with an ampersand (&) as the last nonblank character. The body portion of a UDC contains MPE V/E commands which are executed when the UDC name is entered in place of an MPE V/E command. In addition to MPE V/E commands, you may also include other user-defined commands (nested UDCs) and comment lines. You may not include any other noncommand material such as data for MPE V/E subsystems or user programs.

Several other restrictions also apply. The REDO command is not allowed from within a UDC. If you use DATA, JOB, or HELLO within a UDC, your current job or session is logged off, thereby terminating the UDC execution; no new job or session is logged on. The body may also contain a parameter list which supplies values for the parameters defined in the UDC command header. Refer to "Using Parameters" in this chapter.

---

## Using the SETCATALOG Command

The SETCATALOG command informs MPE V/E that a specified file name contains user-defined commands to be made available to the specified user, account, or system. The command interpreter then searches the file and establishes a directory of all commands contained in the file. The filename is stored in a system catalog of all UDC users (COMMAND.PUB.SYS).

If you want to carry out such a procedure for a file named MYUDC1, for example, enter:

```
:SETCATALOG MYUDC1
```

You may have more than one file to be entered into the directory. In this case, separate the files by commas. For example, enter:

```
:SETCATALOG KEYUDCS,UDCFILE
```

The two files will be entered into the system catalog as a complete and separate set, and will replace any set of UDC files already existing in the catalog.

You cannot use the PURGE command on any UDC file that anyone is accessing. Nor can you use the Editor to modify and keep a file under a file name while anyone whose directory contains a file by the same file name is logged on. Any attempt to do either of these things generates the error message, "EXCLUSIVE VIOLATION".

If you enter the SETCATALOG command with no parameters, all your UDC files are removed from the system UDC catalog. The files are not purged, but are deleted from the catalog. To purge or modify a particular UDC file, you must remove it from the system UDC catalog by issuing the SETCATALOG command with a parameter list that does not include that particular UDC file.

To re-enter UDC files into the catalog, enter the SETCATALOG *filename* command for these files. Refer to the SETCATALOG command in chapter 2.

Removing UDCs with the SETCATALOG command (for example, SETCATALOG;SYSTEM to remove system-wide UDCs) has no immediate effect upon jobs and sessions that are still logged on, except for the job or session that enters the SETCATALOG command. Another job or session may continue to access the canceled UDCs until that job or session terminates, or until each job or session forces the creation of a new UDC directory with a SETCATALOG command.

### 3-6 User-Defined Commands



If you do not have **READ** and **LOCK** access to a UDC file, you cannot use the **SETCATALOG** command on that file, or log on while the file remains active in the system catalog. Therefore, security for the account and group in which the file resides must allow **READ** and **LOCK** access to all users (**R, L; ANY**) at the account and system level. The default file access for the **PUB** group only gives **READ** access to users with **AL, GL**, or home group (**GU**) capability. To override the default, you should use the command **RELEASE filename**, where *filename* is the system-level UDC file. Refer to the **RELEASE** command in chapter 2.

If the UDC file has a lockword associated with it, that lockword may be specified when using the **SETCATALOG** command, as in the following example:

```
:SETCATALOG file1/lockwd1
```

The file and its associated lockword reside in **COMMAND.PUB.SYS**, so you need not know the lockword to log on or to access the UDC. If you fail to specify the lockword when using the **SETCATALOG** command, MPE V/E prompts you for it. If you respond with an invalid lockword, the file is not entered into **COMMAND.PUB.SYS**.

The **USER** option of the **SETCATALOG** command allows those with account manager (**AM**) capability to assign UDC catalogs to any user in their account. It also allows those with system manager (**SM**) capability to assign UDC catalogs to any user in the system. Changes take effect the next time the user logs on.

If the **SETCATALOG** command is part of the user-defined command, it is the last command executed in the UDC body; any subsequent commands in the UDC definition are ignored. Further, if the UDC performing the **SETCATALOG** is a nested UDC (that is, invoked by another UDC), all levels of UDC execution are terminated, and you are returned to the command interpreter.

If you execute a UDC that invokes a program or subsystem and you press **(BREAK)** while the program is running, you may execute a **SETCATALOG** command during the **BREAK**, with a direct command or through another UDC. If you execute a **RESUME** command, the program continues from the point at which the break occurred. However, the UDC will not execute subsequent program or subsystem calls.

For example, suppose you have a UDC which first calls the Editor subsystem, then performs a **SHOWOUT** when you exit the Editor. You invoke the UDC and enter the Editor. While in the Editor, you press **(BREAK)** and enter the **SETCATALOG** command to disassociate your UDC file from your account. The **RESUME** command returns you to the Editor, but the **SHOWOUT** listing is not displayed when you exit.

The system UDC catalog, **COMMAND.PUB.SYS**, must exist for the **SETCATALOG** command to execute properly. If this file does not exist on your system, the system manager or supervisor must build it. The file should normally be built as a binary file with a record size of 20 words. The maximum number of records in the file **COMMAND.PUB.SYS** is determined by the sum of:

- the number of usernames with user-level UDCs; plus
- the number of UDC catalog files specified for each user; plus
- the number of account-level UDCs; plus
- the number of UDC catalog files for each account; plus
- one, if system-level UDCs are in effect; plus
- the number of UDC catalog files specified for system UDCs.

The system manager may build a file for the system catalog in the following way:

```
:BUILD COMMAND.PUB;REC=20,32;DISC=2000
```

To secure `COMMAND.PUB.SYS` so that all system users may have UDCs, but only the system manager may read or modify `COMMAND.PUB.SYS`, the system manager enters:

```
:ALTSEC COMMAND.PUB; (X:ANY;R,L,W,A:CR)
```

This provides EXECUTE (X) access to all users (ANY); READ (R), LOCK (L), WRITE (W), and APPEND (A) access are available only to the creating user (CR), in this case the system manager. For more information on file security specifications, refer to the ALTSEC command in chapter 2 of this manual.

---

## Building and Modifying a UDC File

Once you have decided upon a command, or set of commands, that you would like to save, you may enter them into an ASCII disk file using the Editor. This file has a default record length of 80 bytes for numbered files (standard Editor file), or 72 bytes for unnumbered files. If a line is longer than 72 bytes, the system ignores all data in the additional bytes. You may create several such files, each consisting of one or more user-defined commands. If a file contains more than one user-defined command, the commands must be separated from each other by one line containing one or more asterisks, starting with an asterisk in the first column. The following example shows how to create user-defined commands with the Editor:

```
:EDITOR
HP32201A.7.12 EDIT/3000 MON, AUG 7, 1989, 3:36 PM
(C) HEWLETT-PACKARD CO. 1982
/ADD
1 S ** Header **
2 SHOWJOB ** List of Commands **
3 //
...
/KEEP UDC1
/DELETE ALL
/ADD
1 L
2 LISTF
3 **** ** Separates UDCs **
4 R
5 RESUME
6 //
...
/KEEP UDC2
/EXIT

END OF SUBSYSTEM
:SETCATALOG UDC1,UDC2
```

After setting the catalog, log off. Then log back on. Now you can use the UDCs you created for SHOWJOB, LISTF, and RESUME.

The next example shows how to use “S” to invoke SHOWJOB:

```
:S
_

JOBNUM STATE IPRI JIN JLIST INTRODUCED JOB NAME

#S55 EXEC 20 20 MON 2:49P FIELD.SUPPORT
#S57 EXEC 52 52 MON 2:50P BOB.DATAMGT

2 JOBS:
 0 INTRO
 0 WAIT; INCL 0 DEFERRED
 2 EXEC; INCL 2 SESSIONS
 0 SUSP
JOBFENCE= 2; JLIMIT= 2; SLIMIT= 16
:
```

You can use the Editor to modify commands stored in a UDC file if that file is not being accessed. However, once a UDC file has been entered into the system UDC catalog with the SETCATALOG command, it cannot be modified and kept under the same name while any user whose directory contains that UDC file is logged on.



For example, suppose you are the only user with the file UDC1 (as defined above) as your UDC file. To modify UDC1, enter the following:

```
:SHOWCATALOG

UDC1.SMR.0SE
 S USER
:EDITOR

HP32201A.7.12 EDIT/3000 MON, AUG 7, 1989, 3:36 PM
(C) HEWLETT-PACKARD CO. 1982
/T UDC1
/A 1.1
 1.1 OPTION NOBREAK
 1.2 //
...

/K UDC1
--F-I-L-E---I-N-F-O-R-M-A-T-I-O-N---D-I-S-P-L-A-Y--
| ERROR NUMBER: 90 RESIDUE: 0 |
BLOCK NUMBER: 0 NUMREC: 0
*41*FAILURE TO OPEN KEEP FILE (90)
EXCLUSIVE VIOLATION: FILE BEING ACCESSED (FSERR 90)
/
** BREAK pressed here. **
:SETCATALOG ** File disassociated from system UDC catalog. **
:RESUME
READ PENDING
K UDC1
UDC1 ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD?Y
/E
END OF SUBSYSTEM
:SETCATALOG UDC1
```

---

## Details of UDC Operation

UDCs are initialized at logon, provided that the user, account, or system-level UDCs exist and have been associated by the `SETCATALOG` command, or whenever the user executes a `SETCATALOG` command. The system file, `COMMAND.PUB.SYS`, holds the UDC filenames associated with each user, account, and system UDC, and includes pointers to the files in which the UDCs are defined. At initialization time, all the UDC files you need are opened, and a UDC directory is created for you.

The position of one UDC in the directory determines which other UDCs can call it and which others it can call. The UDC hierarchy imposed by the system depends on how the UDCs in the directory are found in the file. The files are opened and scanned in the order they appeared in the last `SETCATALOG` command for that UDC level. If, for example, the UDC files `CAT1` and `CAT2` are enabled with `SETCATALOG CAT1, CAT2`, then any UDCs in `CAT1` may call any UDCs in `CAT2`. The reverse is not true: UDCs in `CAT2` cannot call UDCs in `CAT1`. All user UDCs are entered into the directory first, then all account UDCs, and all system UDCs are entered last.

If the UDC file has a lockword, you do not need to know the lockword to logon or to access the UDC, since the lockword is kept in `COMMAND.PUB.SYS`. The system manager can release `COMMAND.PUB.SYS` so that all system users may access and execute the UDCs. Only the system manager may read or modify the `COMMAND.PUB.SYS` by entering the `ALTSEC` command. For more information, refer to “Using the `SETCATALOG` Command” in this chapter.

When you are in the command interpreter (CI) and have a colon prompt (`:`), or a remote prompt (`#`), you can execute a UDC by entering the command name, followed by a parameter list (if needed). The UDC directory is scanned for that command, starting at the beginning of the directory. If the command is found, the UDC body for that command is read one line at a time, and parameters are substituted into the line where appropriate.

The CI is re-entered at a special internal entry point. This UDC entry is treated as a new command string, and is handled much the same as all other command strings: the UDC directory is scanned for a UDC that matches the command string. This time, however, the UDC directory scan begins at the directory entry that follows the UDC currently being executed. In this way, UDCs can call other UDCs as long as the called UDC is defined after the UDC being executed. If the command interpreter fails to find a match for the command string in the UDC directory, it tries to match the command string with an MPE V/E command. This implementation allows some UDCs to call other UDCs, but does not allow any kind of UDC recursion. This linear scan of the UDC directory is the heart of UDC invocation. This particular scan allows UDCs with the same name to be executed at different times.

The names of UDCs may be composed of alphanumeric characters, but must begin with an alphabetic character. MPE V/E commands, however, contain only alphabetic characters. A string such as `SHOWDEV6` is treated first as a UDC, because the last character is numeric. If no UDC exists under that name, the MPE V/E command `SHOWDEV` is passed to the parameter. Normal MPE V/E command syntax requires a space between the command and the parameter. The coincidence of names in this example is unusual and produces unexpected results: “`SHOWDEV`” is a valid MPE V/E command, and MPE V/E is capable of separating the command from the parameter and executing them accordingly.

---

## Nesting UDCs

User-defined commands can be nested; that is, they can be defined so that entering one UDC causes several UDCs to execute. For example:

```
DOALL
L
S

L
LISTF

S
SHOWJOB
```

If the preceding set of UDCs is entered into a UDC directory, then each time you enter **DOALL**, the MPE V/E commands **LISTF** and **SHOWJOB** will execute.

A UDC can call another UDC only if the UDC being called is defined after the calling UDC.

If, in the example, **DOALL** is put in the UDC file after the **L** and **S** user-defined commands, the UDC **DOALL** does not execute, and an error message results:

```
L
LISTF

S
SHOWJOB

DOALL
L
S
```

When the command **DOALL** is entered, the UDCs that it calls are not found:

```
:DOALL
L
~
UNKNOWN COMMAND NAME. (CIERR 975)
:
```

System-level UDCs are global to account-level UDCs, which in turn are global to user-level UDCs. This means that an account-level UDC can call a system-level UDC, and a user-level UDC can call either account or system-level UDCs.

If a system and a user-level UDC have the same name, and an account-level UDC calls a UDC by this name, the system-level UDC is called instead of the user-level UDC, because the system-level UDC is global to the user-level UDC.

---

## Errors in a UDC

A UDC contains an error if the execution of an MPE V/E command or another UDC within the body of the UDC results in an error; or if the expansion of the UDC parameters results in an error. If an error or warning occurs as a result of executing a UDC, MPE V/E:

1. Prints the appropriate error message, always.
2. Prints a caret (^) pointing to the error, unless the NOHELP option is specified in the UDC.
3. Prints the line in which the error occurs, unless the NOHELP and NOLIST options are specified.

Every time you log on, the system initializes your UDC directory, as discussed earlier in this chapter. If an illegal UDC is encountered during initialization, or if the UDC file cannot be opened (perhaps it has been purged), only the UDC level in which the error occurs will fail to be initialized. If, for example, the user-level UDC file encounters an error at logon, you still have access to account and system-level UDCs. Similarly, if an account-level UDC encounters an error at logon, you still have access to user and system-level UDCs.

If you anticipate difficulty from a particular command line, you may insert a **CONTINUE** command in the line immediately preceding the potentially troublesome line.

The **CONTINUE** command permits the MPE V/E to bypass the error and proceed with subsequent command lines. In a job, a UDC that generates an error terminates the job, unless the UDC contains a **CONTINUE** command at one of the current nesting levels.

Consider the following UDC file:

```
A
LISTF A@
B
SHOWJOB

B
BUILD FILEX;CODE=XXXX
SHOWTIME

```

Suppose the UDC A is executed:

```
:A
_

FILENAME

APPA APPB APPB1VE APPB2 APPB2VE APPB3
APPB3VE

BUILD FILEX;CODE=XXXX
 ^
UNKNOWN FILE CODE TYPE. (CIERR 253)
```

UDC B contains an error because the BUILD command contains an invalid file code. Therefore, the UDC B terminates and the SHOWTIME following the BUILD command is not executed. Because the execution of UDC B results in an error, SHOWJOB in UDC A is not executed.

A CONTINUE command may be placed before the FILE command in UDC B to allow both UDCs to execute, as follows:

```
A
LISTF A@
B
SHOWJOB

B
CONTINUE
BUILD FILEX;CODE=XXXX
SHOWTIME

```

Suppose the UDC A is executed:

```
:A
_

FILENAME

APPA APPB APPB1VE APPB2 APPB2VE APPB3
APPB3VE

BUILD FILEX;CODE=XXXX
^
UNKNOWN FILE CODE TYPE. (CIERR 253)
MON, AUG 7, 1989, 7:39 AM

JOBNUM STATE IPRI JIN JLIST INTRODUCED JOB NAME

#S118 EXEC 20 20 MON 9:02P OPERATOR.SYS
#S48 EXEC QUIET 223 223 MON 1:51P TEXT.USER
#S69 EXEC 179 179 MON 2:31P SUPPORT.DOC

3 JOBS:
 0 INTRO
 0 WAIT; INCL 0 DEFERRED
 1 EXEC; INCL 3 SESSIONS
 0 SUSP
JOBFENCE= 2; JLIMIT= 5; SLIMIT= 40
```

---

## Using Parameters

The name given to a parameter in the header of a UDC is the formal name by which the parameter is known to the commands in the body of the UDC. "Required" parameters are those for which no default values are supplied in the command header. If a value for a required parameter is not supplied in the parameter list, an error is reported, and the UDC is not executed. If there are no required parameters in the UDC header, a parameter list need not be supplied when the command is executed.

### Types of Parameters

A "keyword" parameter is one in which the formal name of the parameter appears in the parameter list exactly as it appears in the parameter declaration. The command interpreter makes no distinction, though, between upper and lower case alphabetic characters. A "keyword" parameter may appear in any position or in any order in the parameter list.

A "positional" parameter requires the substitution of a name or value in the parameter list in place of the formal name in the parameter declaration. Positional parameters must appear in the parameter list in the same relative position occupied by the formal name in the parameter declaration.

When supplying values for UDC parameters you can use either keyword parameters or positional parameters, but not both at the same time. MPE V/E permits both decimal and octal numbers as command parameters. Octal numbers are preceded by a percent sign (%). The following are examples of different parameter lists for a UDC named Z, with this header:

```
Z PARM1,PARM2=FTN06,PARM3="REC=40,16,F,ASCII"
```

EXAMPLE 1:

```
:Z FTN05
```

The required parameter PARM1 takes the value of FTN05; PARM2 defaults to FTN06, and PARM3 defaults to REC=40,16,F,ASCII.

EXAMPLE 2:

```
:Z FTN05,FTN77
```

PARM1 takes the value FTN05 and PARM2 the value FTN77. The default value for PARM2 is not used because the call contains a new value. The default value (REC=40,16,F,ASCII) is used for PARM3 because PARM3 was not specified. Both FTN05 and FTN77 are positional parameters because they substitute for PARM1 and PARM2.

Example 3 illustrates entering the command Z with keyword parameters. This call is identical in its effect to the call in Example 2. PARM3 takes its default value (REC=40, 16,F,ASCII).

```
:Z PARM2=FTN77,PARM1=FTN05
```

Example 4 illustrates the use of double quotation marks ( " ") to delimit a parameter when that parameter contains embedded blanks or special characters. The second parameter is not specified ( its position is denoted by the adjacent commas), thus the default value FTN06 is used for PARM2, and (REC=-88,16,F,ASCII) is used for PARM3.

```
:Z FTN05,, "REC=-88"
```

The body of the UDC follows the header and specifies which MPE V/E commands the UDC Z is to execute:

```
FILE INPUT=!PARM1,OLD
FILE LISTENTITY =!PARM2,NEW;!PARM3;DISC=1000
RUN ABC
```

Thus, the complete UDC Z is:

```
Z PARM1,PARM2=FTN06,PARM3="REC=-40,16,F,ASCII" ** Header **
FILE INPUT=!PARM1,OLD ** Body **
FILE LISTENTITY =!PARM2,NEW;!PARM3;DISC=1000
RUN ABC
```

## Parameter Delimiters

Placing an exclamation mark (!) before a formal parameter name in the body of the command instructs MPE V/E to substitute the value of that parameter into the command. If a parameter name does not follow the exclamation point, then the string is retained literally, with no error reported to the user.

The first line of the UDC body tells MPE V/E that !PARM1 will take the value of an input file to be specified later:

```
FILE INPUT=!PARM1,OLD
```

Similarly, the second line of the UDC body informs MPE V/E that you wish to substitute an output file called FTN06 (a default value in this example) for !PARM2:

```
FILE LISTENTITY =!PARM2,NEW;!PARM3;DISC=1000
```

The command body of UDC Z contains only the MPE V/E commands FILE and RUN, and parameters for these commands:

```
FILE INPUT=!PARM1,OLD ** Body of UDC Z **
FILE LISTENTITY =!PARM2,NEW;!PARM3;DISC=1000
RUN ABC
```

In the UDC, PARM1 is a required parameter (it has no default value), so its value must be supplied in the parameter list. The parameters PARM2 and PARM3, however, do have specified default values and so are optional. They are passed to the command body either as default values, or as values you enter in the parameter list with the commands:

```
Z PARM1,PARM2=FTN06,PARM3="REC=40,16,F,ASCII"
FILE INPUT=!PARM1,OLD
FILE LISTENTITY =!PARM2,NEW;!PARM3;DISC=1000
RUN ABC
```

The parameters DISC=1000 (for the FILE LISTFILE command) and ABC (for the RUN command) cannot be changed by the user-entered command call.

An exclamation mark in a UDC normally signifies a parameter name. There may be occasions when you want to retain the exclamation mark without signifying a parameter name. Doubling the exclamation mark (!! ) prevents MPE V/E from treating the characters that follow as a parameter name. An odd number of exclamation marks preceding a set of characters causes MPE V/E to expect a value to substitute for the character set. An even

number of exclamation marks preceding a set of characters tells MPE V/E that the character set is not a parameter and tells it not to attempt a value substitution:

```
TESTUDC PARM
OPTION LIST
COMMENT"!!PARM"
COMMENT"!!!PARM"
```

If you execute this UDC with TESTUDC ABC, the two comments are printed as:

```
COMMENT "!!PARM"
COMMENT "!!ABC"
```

You can delimit the UDC parameter by the use of double quotes (" "), and thus insert the parameter into the middle of another string. The quotes are not part of the command, although other blank or nonblank characters within the quotation marks may be. Imagine, for example, having an application program to compile and prepare daily:

| SOURCE   | USL      | PROGRAM |
|----------|----------|---------|
| APPL1SRC | APPL1USL | APPL1   |

You could easily create a UDC to combine the steps to prepare any of the above three source files into a program file (note the commented delimiters on the parameters):

```
COMPAPPL MODULE
OPTION LIST
PURGE APPL!MODULE ** Blank **
SPL APPL!"MODULE"SRC, APPL!"MODULE"USL ** Quotes **
PREP APPL!"MODULE"USL, APPL!MODULE; PMAP ** Semicolon **
SAVE APPL!MODULE ** Blank **
```

After adding the COMPAPPL UDC to the catalog, you may compile and prepare APPL1SRC by entering:

```
:COMPAPPL 1
```

Entering this UDC is the same as entering the following MPE V/E commands:

```
:PURGE APPL1
:SPL APPL1SRC, APPL1USL
:PREP APPL1USL, APPL1; PMAP
:SAVE APPL1
```



---

## Increasing UDC Functionality

UDCs can perform functions considerably more complex than those already covered in this chapter when you include sophisticated command sequences in the body of the UDC. In the following example, the UDC Y2 contains the command sequence normally entered to initiate a remote session:

```
Y2 WHO="USER/PASS.ACCT"
DSLIN Y2
REMOTE HELLO !WHO ** Remote logon to Y2. **
REMOTE
```

Using UDCs in this way can save you having to type frequently used command sequences. Using regular characters, control characters and escape sequences in UDCs allows you to build and control inverse video fields on the terminal. Soft-key and terminal key functions can also be manipulated with advanced UDCs. The terminal keys and soft-keys can then be dynamic throughout your session. If you use Editor functions, for example, you can establish a UDC to program the soft-keys to execute Editor commands.

---

## Using UDCs with Job Control Words

UDCs may be used with job control words (JCWs) to further increase UDC functionality. UDCs that use JCWs within their structure, for example, are useful for monitoring and maintaining the status of your session. The following example of an advanced UDC expands upon the idea of using UDCs to initiate remote sessions. It incorporates a JCW, Y1JCW, which has been previously set to zero in a logon UDC, to first check whether you already have a remote session.

```
CHECKY1 WHO ="USER.ACCT" ** Check on current value of YJCW1. **
IF Y1JCW = 0 THEN
 DSLIN Y1 ** Remote logon. **
 REMOTE HELLO !WHO

 SETJCW Y1JCW = 1 ** Y1CW is set to 1. **
ENDIF

```

If you later try to initiate a remote session on Y1 using the above UDCs, the value of the JCW Y1JCW will be 1. The IF block does not execute, and no new session is initiated.

For more information on job control words, refer to the SETJCW and SHOWJCW commands in chapter 2 of this manual, as well as the *MPE V Intrinsics Reference Manual* (32002-90018).

**Section Divider**

**A. Terminals Supported by MPE V**



## Terminals Supported by MPE V

| Terminal Types | Description                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------|
| 6              | HP 2762A/B (General Electric Terminet 300 or 1200), or Data Communications terminal, Model B (10/15/30/120 cps).             |
| 9              | HP 2165A Terminal (Beehive MiniBee) (10/15/30/60/120/240 cps). Also used for general non-HP terminals.                       |
| 10             | HP 2640A/B, HP 2641A, HP 2644A or HP 2645A. Character mode or full program control of block mode transmission (10 -240 cps). |
| 12             | HP 2645K Katakana/Roman Data Terminal.                                                                                       |
| 13             | Message switching network or other computer.                                                                                 |
| 14             | Multipoint Terminal. Not used for ATP or ADCC ports, only used with MTS/3000.                                                |
| 15             | HP 2635A Printing Terminal. 8-bit protocol (for second character set).                                                       |
| 16             | HP 2635A Printing Terminal. 7-bit protocol (standard character set).                                                         |
| 18             | General non-HP terminal. No ENQ/ACK handshaking. Some application printers use this <i>termtype</i> (e.g. 2601A).            |
| 19             | Serial spooled printers.                                                                                                     |
| 20             | 8-bit spooled serial printer (not supported with subtype 15).                                                                |
| 21             | Spooled serial printer with escape sequences allowed.                                                                        |
| 22             | 8-bit spooled serial printer with embedded escape sequences allowed.                                                         |
| TTPCL18        | HP2687A (MPE V/E G.01.00 or later).                                                                                          |
| TTPCL19        | 7-bit PCL serial spooled sprinters.                                                                                          |
| TTPCL22        | 8-bit PCL serial spooled sprinters.                                                                                          |

For further information refer to the Fundamental Communications Handbook (5957-4634).





**Section Divider**

**B. Subsystem Formal File Designators**



## Subsystem Formal File Designators

---

| Command     | Parameter                                                                                   | Formal File Designator                            |
|-------------|---------------------------------------------------------------------------------------------|---------------------------------------------------|
| :BASIC      | <i>commandfile</i><br><i>inputfile</i><br><i>listfile</i>                                   | BASCOM<br>BASIN<br>BASLIST                        |
| :BASICOMP   | <i>textfile</i><br><i>uslfile</i><br><i>listfile</i>                                        | BSCTEXT<br>BSCUSL<br>BSCLIST                      |
| :BASICGO    | <i>textfile</i><br><i>listfile</i>                                                          | BSCTEXT<br>BSCLIST                                |
| :BASICPREP  | <i>textfile</i><br><i>listfile</i>                                                          | BSCTEXT<br>BSCLIST                                |
| :BBASIC     | <i>commandfile</i><br><i>inputfile</i><br><i>listfile</i>                                   | BASCOM<br>BASIN<br>BASOUT                         |
| :BBASICOMP  | <i>infile</i><br><i>uslfile</i><br><i>listfile</i>                                          | BBCIN<br>BBCUSL<br>BBCLIST                        |
| :BBASICGO   | <i>infile</i><br><i>listfile</i>                                                            | BBCIN<br>BBCLIST                                  |
| :BBASICPREP | <i>infile</i><br><i>listfile</i>                                                            | BBCIN<br>BBCLIST                                  |
| :COBOL      | <i>textfile</i><br><i>uslfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i> | COBTEXT<br>COBUSL<br>COBLIST<br>COBMAST<br>COBNEW |
| :COBOLGO    | <i>textfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i>                   | COBTEXT<br>COBLIST<br>COBMAST<br>COBNEW           |
| :COBOLPREP  | <i>textfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i>                   | COBTEXT<br>COBLIST<br>COBMAST<br>COBNEW           |



| Command                                                                                      | Parameter                                                                                                            | Formal File Designator                                        |
|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| :COBOLII                                                                                     | <i>textfile</i><br><i>uslfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i><br><i>workspace</i>      | COBTEXT<br>COBUSL<br>COBLIST<br>COBNEW<br>COBWKSP             |
| :COBOLIIGO                                                                                   | <i>textfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i><br><i>workspacename</i>                    | COBTEXT<br>COBLIST<br>COBMAST<br>COBNEW<br>COBWKSP            |
| :COBOLIIPREP                                                                                 | <i>textfile</i><br><i>progfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i><br><i>workspacename</i> | COBTEXT<br>COBPROG<br>COBLIST<br>COBMAST<br>COBNEW<br>COBWKSP |
| :EDITOR                                                                                      | <i>listfile</i>                                                                                                      | EDTLIST                                                       |
| :FORTRAN                                                                                     | <i>textfile</i><br><i>uslfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i>                          | FTNTEXT<br>FTNUSL<br>FTNLIST<br>FTNMAST<br>FTNNEW             |
| :FORTGO                                                                                      | <i>textfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i>                                            | FTNTEXT<br>FTNLIST<br>FTNMAST<br>FTNNEW                       |
| :FORTPREP                                                                                    | <i>textfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i>                                            | FTNTEXT<br>FTNLIST<br>FTNMAST<br>FTNNEW                       |
| :FTN77 (FORTRAN 77)   <i>textfile</i><br><i>uslfile listfile</i>   FTNTEXT FTNUSL<br>FTNLIST |                                                                                                                      |                                                               |
| :FTNGO                                                                                       | <i>textfile</i><br><i>listfile</i>                                                                                   | FTNTEXT<br>FTNLIST                                            |
| :FTNPREP                                                                                     | <i>textfile</i><br><i>progfile</i><br><i>listfile</i>                                                                | FTNTEXT<br>FTNPROG<br>FTNLIST                                 |
| :PASCAL                                                                                      | <i>textfile</i><br><i>uslfile</i><br><i>listfile</i>                                                                 | PASTEXT<br>PASUSL<br>PASLIST                                  |

## B-2 Subsystem Formal File Designators

| Command         | Parameter                                                                                   | Formal File Designator                            |
|-----------------|---------------------------------------------------------------------------------------------|---------------------------------------------------|
| :PASCALGO       | <i>textfile</i><br><i>listfile</i>                                                          | PASTEXT<br>PASLIST                                |
| :PASCALPREP     | <i>textfile</i><br><i>listfile</i>                                                          | PASTEXT<br>PASLIST                                |
| :PREP :PREPRUN  | PMAP                                                                                        | SEGLIST                                           |
| :PREPRUN :RUN   | LMAP                                                                                        | LOADLIST                                          |
| :RESTORE        | SHOW                                                                                        | SYSLIST                                           |
| :RPG            | <i>textfile</i><br><i>uslfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i> | RPGTEXT<br>RPGLIST<br>RPGMAST<br>RPGNEW           |
| :RPGGO          | <i>textfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i>                   | RPGTEXT<br>RPGLIST<br>RPGMAST<br>RPGNEW           |
| :RPGPREP        | <i>textfile</i><br><i>infile</i><br><i>masterfile</i><br><i>newfile</i>                     | RPGTEXT<br>RPGLIST<br>RPGMAST<br>RPGNEW           |
| :SEGMENTER      | <i>listfile</i>                                                                             | SEGLIST                                           |
| :SPL            | <i>textfile</i><br><i>uslfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i> | SPLTEXT<br>SPLUSL<br>SPLLIST<br>SPLMAST<br>SPLNEW |
| :SPLGO          | <i>textfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i>                   | SPLTEXT<br>SPLLIST<br>SPLMAST<br>SPLNEW           |
| :SPLPREP        | <i>textfile</i><br><i>listfile</i><br><i>masterfile</i><br><i>newfile</i>                   | SPLTEXT<br>SPLLIST<br>SPLMAST<br>SPLNEW           |
| :STORE :RESTORE | SHOW                                                                                        | SYSLIST                                           |
| :VINIT          | <i>listfile</i>                                                                             | VINLIST                                           |



**Section Divider**

**Index**





## Index

---

### A

#### Abort

- I/O request, 2-12
- Job or session (=ABORTJOB), 2-14
- Job or session (ABORTJOB), 2-14
- Pending I/O request (=ABORTIO), 2-12
- Pending I/O request (ABORTIO), 2-12
- Program/operation (ABORT), 2-12
- Scheduled job, 2-15
- Waiting job, 2-15

#### ABORT

- Abort a program/operation, 2-12
- Initiating stack dump facility, 2-312

#### Aborting a program, 1-8

#### Aborting jobs/sessions, 2-217

#### =ABORTIO

- Control A, 2-12
- Used when ABORTIO ineffective, 2-13

#### ABORTIO

- Abort pending I/O request, 2-12
- Not effective - use =ABORTIO, 2-13

#### =ABORTJOB

- Control A, 2-14

#### ABORTJOB

- Abort a job or session, 2-14

#### Abort pending I/O request

- Device, 2-12

#### ACCEPT

- Permit job/session/data acceptance, 2-16

#### Access

- Granted to system operator command (ALLOW), 2-20

#### Accessing files, 2-132

#### Accessing the online help facility, 2-166

#### Account attributes, 2-221

- Changing, 2-22

#### Account capabilities, 2-222

#### Account Capabilities, 2-24

#### Account library, 2-250

#### Account manager

- definition, 1-2

#### Accounts

- Creating, 2-221
- Deleting, 2-256
- Displaying information, 2-183, 2-274
- Resetting connect-time, 2-278

#### Resetting CPU-time, 2-278

#### Acquiring global RINs, 2-156

#### ADCs

- associated with I/O devices, 2-330

#### Allocate

- Memory, 2-18

#### ALLOCATE

- Caution, 2-19
- DEALLOCATE, 2-18
- Permanently load a program/procedure, 2-18

#### Allocation of programs

- Displaying status, 2-320

#### ALLOW

- Grant access to system operator command, 2-20

#### SHOWALLOW, 2-22

#### ALLOW command, 1-3

#### Allowing

- Operator control of a device to a user, 2-48

#### ALTACCT

- Caution, 2-25

#### ALTACCT command, 2-22

#### Alter attributes

- Account (ALTACCT), 2-22

#### Altering

- File security, 2-36
- Job attributes, 2-32
- printing priority, 2-40
- Spoolfile characteristics, 2-40
- User attributes, 2-42
- User logging identifier attributes, 2-34

#### Alternate command sequences

- Executing, 2-115

#### Alternative listing devices

- jobs, 2-388

#### ALTGROUP command, 2-28

#### ALTJOB command, 2-32

#### ALTLOG command, 2-34

#### ALTSEC command, 2-36

#### ALTSPoolFILE command, 2-40

#### ALTVSET command, 2-46

#### Ampersand (&)

- continuation character, 1-5

#### ASCII, 2-128

#### ASCII files

- Creating, 2-114

#### ASOCTBL.PUB.SYS, 2-48

- Assigning job numbers, 2-171
- ASSOCIATE command, 1-3, 2-48
- Attributes
  - Account, 2-22, 2-221
  - File, 2-68, 2-123
  - Group, 2-28
  - job, 2-32
  - User, 2-42
  - User logging identifiers, 2-34
- AUTOALLOCATE command, 2-50
- Automatic allocation/deallocation of programs, 2-50
- Automatic log file changing, 2-78

## B

- Backing up the system, 2-153, 2-379, 2-395
- Backup
  - partial, 2-236
- BASIC command, 2-51
- BASICGO command, 2-53
- BASICOMP command, 2-55
- BASICPREP command, 2-57
- BASIC/V programs
  - Compiling, 2-53, 2-55, 2-57
  - Executing, 2-53
  - Interpreting, 2-51
  - Preparing, 2-53, 2-57
- Batch jobs
  - Creating, 2-171
  - Ending, 2-120
  - spooling, 2-385
  - Starting, 2-171
- Baud rate, terminals, 2-361
- BBASIC command, 2-59
- BBASICGO command, 2-61
- BBASICOMP command, 2-63
- BBASICPREP command, 2-65
- Binary, 2-128
- Blockfactor
  - Files, 2-128
- Block screen mode
  - overriding, 2-411
- BREAK, 1-7
  - disabling, 1-8
- Breakable commands
  - nonprogram, 1-7
  - program, 1-7
- BREAKJOB command, 2-67
- Buffering files, 2-132
- Business BASIC/V programs
  - Compiling, 2-59, 2-61, 2-63, 2-65
  - Executing, 2-59, 2-61
  - Preparing, 2-59, 2-61, 2-65

## C

- CACHECONTROL command, 2-76
- Caching on a single disk, 2-369, 2-376
- Cancelling file equations, 2-277
- Cancelling volume sets, 2-180
- Capabilities
  - Account, 2-24, 2-222
  - Special, 2-1
  - System manager/supervisor, 2-76
  - User, 2-22, 2-42
- Capabilities, special, 1-2
- Capability
  - System Supervisor, 2-50
- Caret (^)
  - meaning of, 1-6
- Carriage-control characters, 2-132
- Catalogs
  - activating and deactivating, 2-309
- Caution
  - ALLOCATE, 2-19
  - ALTACCT, 2-25
- CHANGELOG command, 2-78
- Changing
  - Account attributes, 2-22
  - File security, 2-36
  - filters of circular subqueues, 2-401
  - Group attributes, 2-28
  - Job attributes, 2-32
  - Log files automatically, 2-78
  - Logging files, 2-78
  - priority limits of circular subqueues, 2-401
  - User logging identifier attributes, 2-34
- Changing execution priority for jobs, 2-178
- Changing file identities, 2-270
- Changing log files, 2-394
- CHanging passwords, 2-244
- Circular subqueues
  - changing filters/priority limits, 2-401
- CIR files, 2-131
- Closing spool queues, 2-359
- COBOL command, 2-80
- COBOLGO command, 2-82
- COBOLII command, 2-86
- COBOLIIIGO command, 2-88
- COBOLIIPREP command, 2-90
- COBOLII programs
  - Compiling, 2-86, 2-88, 2-90
  - Executing, 2-88
  - Preparing, 2-88, 2-90
- COBOLPREP command, 2-84
- COBOL programs
  - Compiling, 2-80, 2-82, 2-84
  - Executing, 2-82
  - Preparing, 2-82, 2-84

- Codes
  - Files, 2-131
- Command
  - logging on to system, 2-8
- Command elements, 1-3
- Command entry
  - Description, 2-1
- Command files
  - Controlling execution, 2-169
- Command interpreter, 1-5
  - Using, 2-162
- Command length, 1-3
- Command lines
  - Editing, 2-263
  - Reexecuting, 2-263
- COMMAND LOGON, 2-8
- Commands
  - breakable nonprogram, 1-7
  - console, 1-3
  - Control-A, 2-1
  - entering, 1-5
  - Executing alternate sequences, 2-115
  - functions, 1-1
  - interrupting execution, 1-7
  - Listed by function, 2-3
  - nonbreakable nonprogram, 1-7
  - nonprogram, 1-6
  - operator, 1-2, 2-323
  - program, 1-6
  - program (breakable), 1-7
  - Table of, 2-3
- Command stream
  - Inserting comments, 2-92
- COMMENT command, 2-92
- Communication devices
  - Displaying status, 2-328
- Compiling
  - BASIC/V programs, 2-53, 2-55, 2-57
  - Business BASIC/V programs, 2-59, 2-61, 2-63, 2-65
  - COBOLII programs, 2-86, 2-88, 2-90
  - COBOL programs, 2-80, 2-82, 2-84
  - FORTRAN/3000 programs, 2-140, 2-142, 2-144
  - FORTRAN 77 programs, 2-149, 2-151
  - SPL/3000 programs, 2-362, 2-364, 2-366
- Compiling FORTRAN 77 programs, 2-147
- Compiling Pascal/3000 programs, 2-238, 2-240, 2-242
- Compiling RPG programs, 2-293, 2-295, 2-297
- Configuration dialog
  - starting, 2-395
- Connect-time
  - Resetting, 2-278
- Connect-time setting, 2-161
- Console
  - Changing to another terminal, 2-93
  - Displaying logical device number, 2-93
  - moving, 1-3
  - Moving, 2-93
  - sending messages to, 2-400
- CONSOLE command, 2-93
- Console commands, 1-3
- Console messages, 2-262
- Continuation character
  - ampersand (&), 1-5
- CONTINUE command, 2-95
- Control-A
  - Commands, 2-1
  - With the LOGOFF command, 2-217
  - With the LOGON command, 2-218
- Control A
  - =ABORTIO, 2-12
  - =ABORTJOB, 2-14
- Controlling execution of command files, 2-169
- Controlling execution of jobs, 2-169
- Controlling execution of UDCs, 2-169
- Control maintenance processor messages/prompts, 2-93
- Copying files, 2-121, 2-134
- CPU-time
  - Resetting, 2-278
- CPU-time setting, 2-161
- Creating
  - ASCII files, 2-114
  - Files, 2-68
  - Program files, 2-114
  - sessions on a device, 2-370
  - Text files, 2-114
- Creating accounts, 2-221
- Creating batch jobs, 2-171
- Creating groups, 2-225
- Creating JCWs, 2-314
- Creating log identifiers, 2-154
- Creating new private volume sets, 2-231
- Creating new users, 2-228
- Creating welcome messages, 2-413
- CS subqueues, 2-402

## D

- Data
  - Disabling, 2-265
  - enabling/disabling, 2-391
  - Permit, 2-16
  - spooling, 2-385
- Data-accepting
  - Device, 2-12, 2-16
- DATA command, 2-96
- Date, displaying, 2-356
- Deactivating catalogs, 2-309
- DEALLOCATE



- ALLOCATE, 2-18
- DEALLOCATE command, 2-99
- Deallocation of programs
  - Automatic, 2-50
- DEBUG command, 2-100
- Debug facility
  - Disarming, 2-279
  - Entering, 2-100
- Default settings
  - START NORECOVERY, 2-402
  - START RECOVERY, 2-402
- Default values
  - Files, 2-74
- Defining new private volume sets, 2-231
- Defining new users, 2-228
- Definition of volume sets, 2-46
- DELETESPOOLFILE command, 2-101
- Deleting
  - Ready spool files, 2-101
  - System-defined input spool files, 2-101
- Deleting accounts, 2-256
- Deleting disk files, 2-255
- Deleting groups, 2-258
- Deleting spool files from disk, 2-101
- Deleting \$STDLIST, 2-308
- Deleting users, 2-260
- Deleting volume sets, 2-261
- Description
  - Command entry, 2-1
- Device
  - Data-accepting, 2-12, 2-16
  - Job-accepting, 2-12, 2-16
- Device classes
  - Opening spool queues, 2-233
- Device files
  - Entering data, 2-96
- Devices
  - Allowing users operator control, 2-48
  - Downed, 2-157
  - Opening spool queue, 2-233
  - releasing, 2-397
  - Removing from use, 2-108
  - returning to normal functions, 2-404
  - RPS, enabling/disabling, 2-106
  - spooled, 2-393
  - spooling, 2-374
- Diagnostic control unit messages/prompts, 2-93
- Diagnostics
  - Assigning downed devices, 2-157
- Disable
  - Automatic program allocation, 2-50
- Disabling
  - jobs/data, 2-391
  - RPS devices, 2-106
- Disabling BREAK, 1-8
- Disabling jobs/sessions/data, 2-265
- Disabling message reception, 2-319
- Disabling movable volume facility, 2-408
- Disabling single disk caching, 2-376
- Disabling stack dump facility, 2-312
- DISALLOW command, 2-103
- Disarming
  - Debug facility, 2-279
- DISASSOCIATE command, 2-105
- DISCRPS command, 2-106
- Disk caching, 2-76
  - disabling, 2-376
  - enabling, 2-369
- Disk caching performance
  - summary, 2-324
- Disk drives
  - Types used with ALTVSET, 2-46
- Disk drive status
  - Displaying, 2-112
- Disks
  - Foreign, 2-138
  - Restoring files, 2-280
- DISMOUNT command, 2-107
- Displaying
  - allowed operator commands, 2-323
  - communication device status, 2-328
  - Console logical device number, 2-93
  - current date and/or time, 2-356
  - input device file status, 2-333
  - I/O device status, 2-330
  - job/session status, 2-347
  - process scheduling data, 2-354
  - Status of JCW , 2-337
  - status of jobs/sessions, 2-339
  - status of output device files, 2-349
  - status of program allocations, 2-320
  - subqueue information, 2-354
  - system log file information, 2-344
  - user log file status, 2-345
  - volume set users, 2-410
- Displaying account information, 2-183
- Displaying disk drive status, 2-112
- Displaying file equations, 2-186
- Displaying file information, 2-187
- Displaying group information, 2-203
- Displaying information
  - Accounts, 2-274
  - Groups, 2-274
- Displaying job information, 2-173
- Displaying logging identifiers, 2-206
- Displaying pending console REPLY messages, 2-262
- Displaying temporary file information, 2-195
- Displaying user information, 2-208
- Disposition of files, 2-135

- DOWN command, 2-108
- Downed devices
  - Assigned to diagnostics, 2-157
- DOWNLOAD command, 2-110
- DS subqueues, 2-402
- DSTAT command, 2-112
- E**
- Editing command lines, 2-263
- Editing files, 2-114
- EDITOR command, 2-114
- EDIT subsystem, 2-114
- ELSE command, 2-115
- Enable
  - Automatic allocation/deallocation of programs, 2-50
- Enabling
  - RPS devices, 2-106
- Enabling jobs/data, 2-391
- Enabling jobs/sessions, 2-218
- Enabling message reception, 2-319
- Enabling movable volume facility, 2-408
- Enabling single disk caching, 2-369
- Enabling stack dump facility, 2-312
- ENDIF command, 2-116
- Ending
  - Sessions, 2-75
- Ending batch jobs, 2-120
- End of file
  - Denoting, 2-119
- Entering
  - Data from device files, 2-96
  - Debug facility, 2-100
- Entering commands, 1-5
- Environment files, 2-125
- :EOD command, 2-117
  - Use with the DATA command, 2-117
- :EOF
  - command, 2-119
- EOJ command, 2-120
- Errors
  - Overriding, 2-95
- Errors in command entry, 1-6
- ES subqueues, 2-402
- Establishing passwords, 2-244
- Excluding files from restore process, 2-281
- Executing
  - BASIC/V programs, 2-53
  - Business BASIC/V programs, 2-59, 2-61
  - COBOLII programs, 2-88
  - COBOL programs, 2-82
  - FORTTRAN/3000 programs, 2-140
  - FORTTRAN 77 programs, 2-149
  - SPL/3000 programs, 2-364
- Executing alternate command sequences, 2-115

- Executing jobs
  - Changing priority, 2-178
- Executing nonprogram commands, 1-6
- Executing Pascal/3000 programs, 2-240
- Executing prepared programs, 2-300
- Executing programs, 2-249, 2-300
- Executing RPG programs, 2-295
- Execution priority
  - Jobs, 2-172
- Extents
  - Files, 2-128
- F**
- FCOPY command, 2-121
- Fformat information
  - Downloading, 2-110
- File attributes, 2-68
- FILE command, 2-123
- FILE commands for subsystems; Implicit, 2-136
- File designator
  - Implicit, 2-136
- File domains, 2-125
- File equations, 2-123
  - Cancelling, 2-277
  - Displaying, 2-186
- File information
  - Displaying, 2-187
- File labels, 2-133
- File, log
  - displaying status, 2-345
- Files
  - Accessing, 2-132
  - Altering security, 2-36
  - Blockfactor, 2-128
  - Buffering, 2-132
  - Changing identities, 2-270
  - CIR, 2-131
  - Codes, 2-131
  - Copying, 2-121, 2-134
  - Creating, 2-68
  - Declaring attributes, 2-123
  - Default values, 2-74
  - Deleting, 2-255
  - Disposition of, 2-135
  - Editing, 2-114
  - Environment, 2-125
  - Excluding from restore process, 2-281
  - Extents, 2-128
  - Full backup, 2-153
  - MSG, 2-131
  - NORIO, 2-131
  - partial backup, 2-236
  - Record size, 2-127
  - Recovering, 2-280
  - Removing security provisions, 2-266

- Restoring security provisions, 2-306
- Restoring to disk, 2-280
- Restoring with TurboSTORE, 2-284
- RIO, 2-131
- Saving, 2-304
- Sharing, 2-132
- STD, 2-131
- storing to tape, 2-379
- system-defined, 1-5
- Tape density, 2-128
- Temporary, 2-195
- User-defined command (UDC), 2-326
- USL, 2-246, 2-252
- File security
  - Changing, 2-36
- Files, input device
  - Displaying status, 2-333
- Files, log
  - changing, 2-394
- Files, output device
  - displaying status, 2-349
  - ranking, 2-352
- File, system log
  - displaying information, 2-344
- Filters
  - changing for circular subqueues, 2-401
- FOREIGN command, 2-138
- Foreign Disks
  - formatting, initializing and labeling, 2-405
- Formatting
  - foreign disks, 2-405
  - private volumes, 2-405
  - serial disks, 2-405
- Forms messages, 2-134
- FORTGO command, 2-140
- FORTPREP command, 2-142
- FORTTRAN/3000 programs
  - Compiling, 2-140, 2-142, 2-144
  - Executing, 2-140
  - Preparing, 2-140, 2-142
- FORTTRAN 77 programs
  - Compiling, 2-147, 2-149, 2-151
  - Executing, 2-149
  - Preparing, 2-149, 2-151
- FORTTRAN command, 2-144
- FREERIN command, 2-146
- FTN command, 2-147
- FTNGO command, 2-149
- FTNPREP command, 2-151
- FULLBACKUP command, 2-153

## G

- GETLOG command, 2-154
- GETRIN command, 2-156
- Getting information
  - Online, 2-166
- GIVE command, 2-157
- Global RINs
  - Acquiring, 2-156
  - Releasing, 2-146
- Group attributes
  - Changing, 2-28
- Group library, 2-250
- Group passwords, 2-225
- Groups
  - Creating, 2-225
  - Deleting, 2-258
  - Displaying information, 2-203, 2-274
  - Logon, 2-162
  - Volume sets, 2-226

## H

- Header output
  - Resuming, 2-159
  - Stopping, 2-158
- HEADOFF command, 2-158
- HEADON command, 2-159
- HELLO command, 2-160
- HELP command, 2-166
- HELP facility, 1-8

## I

- IF block
  - Terminating, 2-116
- IF command, 2-169
- Implicit FILE commands for subsystems, 2-136
- Increasing volume sets, 2-46
- Indirect files , 2-281
- Initializing
  - foreign disks, 2-405
  - private volumes, 2-405
  - serial disks, 2-405
- Input device files
  - Displaying status, 2-333
- Input devices
  - Displaying status, 2-330
- Input priority
  - Jobs, 2-161, 2-172
  - Jobs/sessions, 2-176
- Inserting
  - Comments into command stream, 2-92
- Interactive sessions
  - Starting, 2-160
- Interpreting BASIC/V programs, 2-51
- Interrupting command execution, 1-7

## J

### JCWs

- assigning values, 2-314
- creating, 2-314

### Job-accepting

- Device, 2-12, 2-16

### JOB command, 2-171

### Job control words

- Displaying status, 2-337
- System-defined, 2-337
- System-reserved, 2-337
- values, 2-314

### Job execution priority

- Changing, 2-178

### JOBFENCE command, 2-176

### Job files

- sequence numbers, 1-6

### Jobfiles

- Denoting end-of-data, 2-117
- Denoting end-of-file, 2-119

### Job information

- Displaying, 2-173

### Job numbers, 2-171

### JOBPRI command, 2-178

### Jobs

- Aborting, 2-217
  - alternative listing devices, 2-388
  - Changing attributes, 2-32
  - changing execution priority, 2-178
  - Controlling execution, 2-169
  - Disabling, 2-265
  - Displaying status, 2-339
  - Enabling, 2-218
  - enabling/disabling, 2-391
  - Execution priority, 2-172
  - Input priorities, 2-172
  - Input priority, 2-161, 2-176
  - Limiting number of, 2-181
  - Logging on, 2-171
  - Overriding errors, 2-95
  - Priority queues, 2-172
  - Restricting execution, 2-179
  - scheduling, 2-387
  - Security, 2-179
  - sending messages, 2-411
  - spooling, 2-385
  - streaming, 2-387
  - subsystem commands, 2-388
  - Suspending, 2-67
- ### JOBSECURITY command, 2-179
- ### Job status
- displaying, 2-347

## L

### Labeling

- foreign disks, 2-405
- private volumes, 2-405
- serial disks, 2-405

### Labels

- Files, 2-133

### LDISMOUNT command, 2-180

### Length of command, 1-3

### Library program files, 2-250

### LIMIT command, 2-181

### Limiting the number of jobs/sessions, 2-181

### Line printers

- Downloading format information, 2-110

### LISTACCT command, 2-183

### LISTEQ command, 2-186

### LISTF command, 2-187

### LISTFTEMP command, 2-195

### LISTGROUP command, 2-203

### Listing UDC files, 2-326

### Listing volume set definitions, 2-211

### LISTLOG command, 2-206

### LISTUSER command, 2-208

### LISTVS command, 2-211

### L\_MOUNT command, 2-214

### Load

- Procedure , 2-18
- Program , 2-18
- Program/procedure (ALLOCATE), 2-18

### LOG command, 2-216

### Log files

- changing, 2-394
- Naming, 2-78

### Logging

- Resuming, 2-290
- System, 2-206
- User, 2-269

### Logging files

- Changing, 2-78

### Logging identifiers

- Creating, 2-154
- Displaying, 2-206
- User, 2-154

### Logging off, 2-75

### Logging on, 2-160

- Jobs, 2-171

### Logical device number

- displaying, 2-347

### =LOGOFF command, 2-217

### =LOGON command, 2-218

### LOGON Command, 2-8

### Logon groups, 2-162



## M

### Magnetic tape

- Restoring files from, 2-280

### Memory

- Allocate, 2-18
- Deallocating, 2-99

### Messages

- Console, 2-262
- Control maintenance processor, 2-93
- Diagnostic control unit, 2-93
- received by \$STDLIST, 2-319
- REPLY, 2-262
- sending to sessions, 2-398
- sending to the console, 2-400

Messages about forms, 2-134

Minimum printing priority, 2-234

Modifying volume set definitions, 2-46

MOUNT command, 2-219

Movable volume facility

- enabling/disabling, 2-408

Moving the system console, 1-3, 2-93

MSG files, 2-131

## N

### Naming

- Log files, 2-78
- sessions, 2-370

Naming volume sets, 2-408

NEWACCT command, 2-221

NEWGROUP command, 2-225

\$NEWPASS, 2-123

NEWUSER command, 2-228

NEWVSET command, 2-231

Nonbreakable commands, 1-7

NS/3000/XL AdvanceNet, 2-123

\$NULL, 2-123

## O

### Offline devices

- releasing, 2-397

\$OLDPASS, 2-123, 2-304

Opening spool queues, 2-233

OPENQ command, 2-233

### Operation

- Abort, 2-12

- Resuming execution, 2-288

### Operator

- sending messages to, 2-400

Operator commands, 1-2, 2-48

- displaying, 2-323

- Prohibiting access, 2-103

OUTFENCE command, 2-234

### Output

- suspending, 2-393

### Output device files

- displaying status, 2-349
- ranking, 2-352

### Output devices

- Displaying status, 2-330

### Output priority

- Spool files, 2-234

### Output spoolfiles

- Altering characteristics, 2-40

### Overriding

- Job errors, 2-95

Overriding block screen mode, 2-411

Overriding quiet mode, 2-411

## P

### Parameters

- characteristics, 1-4

- keyword groups, 1-4

- positional, 1-4

PARTBACKUP command, 2-236

Pascal/3000 programs

- Compiling, 2-238, 2-240, 2-242

- Executing, 2-240

- Preparing, 2-240, 2-242

PASCAL command, 2-238

PASCALGO command, 2-240

PASCALPREP command, 2-242

PASSWORD command, 2-244

### Passwords

- Account, 2-221

- Changing, 2-244

- Establishing, 2-244

- Group, 2-225

- Users, 2-228

- Using, 2-163

- using with STARTSESS, 2-372

Pending console REPLY messages

- Displaying, 2-262

Pending resource requests

- Replying to, 2-272

### Performance

- Disk caching, 2-76

Performing backups, 2-153

Permanent files, 2-74

- Saving, 2-304

### Permit

- Data, 2-16

- Job, 2-16

- Job/session/data acceptance (ACCEPT), 2-16

- Session, 2-16

Physical end-of-file, 2-119

### PIN

- with labels, 2-355

Prepared programs

- Executing, 2-300

- Preparing
  - BASIC/V programs, 2-53, 2-57
  - Business BASIC/V programs, 2-59, 2-61, 2-65
  - COBOLII programs, 2-88, 2-90
  - COBOL programs, 2-82, 2-84
  - FORTTRAN/3000 programs, 2-140, 2-142
  - FORTTRAN 77 programs, 2-149, 2-151
  - SPL/3000 programs, 2-364, 2-366
- Preparing Pascal/3000 programs, 2-240, 2-242
- Preparing programs, 2-246, 2-249
- Preparing RPG programs, 2-295, 2-297
- PREP command, 2-246
- PREPRUN command, 2-249
- Printing
  - Changing priority, 2-40
- Printing priority, 2-234
- Priority
  - Printing, 2-234
- Priority limits
  - changing for circular subqueues, 2-401
- Priority queues
  - Jobs, 2-172
- Private Volumes
  - formatting, initializing and labeling, 2-405
- Private volume sets
  - defining new, 2-231
- Privileged mode programs, 2-249
- Procedures
  - Deallocating, 2-99
- Processes
  - Abnormal termination, 2-279
- Process identification numbers, 2-355
- Process scheduling data
  - displaying, 2-354
- Program
  - abort, 2-12
- Program allocation
  - Displaying status, 2-320
- Program allocation/deallocation, 2-50
- Program files
  - Creating, 2-114
- Programs
  - Deallocating, 2-99
  - EDIT, 2-114
  - Executing, 2-249, 2-300
  - FCOPY, 2-121
  - In privileged mode, 2-249
  - Preparing, 2-246, 2-249
- Prohibiting access to operator commands, 2-103
- Prompt, 1-4
- Prompts
  - Control maintenance processor, 2-93
  - Diagnostic control unit, 2-93
- PTAPE command, 2-254
- PURGEACCT command, 2-256
- PURGE command, 2-255
- PURGEGROUP command, 2-258
- PURGEUSER command, 2-260
- PURGEVSET command, 2-261
- Q**
  - Queue priority for jobs, 2-171
- Queues
  - spool, 2-233
- Quiet mode, 2-398
  - overriding, 2-411
- R**
  - Rank of output device files, 2-352
- READ errors
  - Disk caching, 2-76
- Reading paper tapes, 2-254
- RECALL command, 2-262
- Record size
  - Files, 2-127
- Recovering files, 2-280
- REDO command, 2-263
- Reexecuting command lines, 2-263
- REFUSE command, 2-265
- RELEASE command, 2-266
- Releasing
  - Table entries, 2-99
  - Volume sets, 2-107
- Releasing devices, 2-397
- Releasing global RINs, 2-146
- Releasing table entries, 2-99
- RELLOG command, 2-269
- Removing
  - User control of devices, 2-105
- Removing devices from use, 2-108
- Removing file security provisions, 2-266
- Removing user logging identifiers, 2-269
- RENAME command, 2-270
- REPLY command, 2-272
- REPLY message, 2-262
- REPORT command, 2-274
- Requesting resources, 2-272
- Reserving volume sets, 2-214, 2-219
- RESETACCT command, 2-278
- RESET command, 2-277
- RESETDUMP command, 2-279
- Resetting account connect-time, 2-278
- Resetting account CPU-time, 2-278
- Resources
  - Requesting, 2-272
  - Restricting access, 2-179
- Restarting user logging, 2-216
- RESTORE command, 2-280
- Restore process
  - Excluding files from, 2-281

- Restoring files to disk, 2-280
- Restoring security provisions, 2-306
- Restricting execution of jobs, 2-179
- Restricting resource access, 2-179
- RESUME command, 2-288
- RESUMEJOB command, 2-289
- RESUMELOG command, 2-290
- RESUMESPOOL command, 2-291
- Resuming execution of an operation, 2-288
- Resuming header/trailer output, 2-159
- Resuming spooler output, 2-291
- Resuming suspended jobs, 2-289
- Resuming system logging, 2-290
- Returning devices to normal functions, 2-404
- RINs
  - Acquiring, 2-156
- RIO files, 2-131
- Rotational position sensing, 2-106
- RPG command, 2-293
- RPGGO command, 2-295
- RPGPREP command, 2-297
- RPG programs
  - Compiling, 2-293, 2-295, 2-297
  - Executing, 2-295
  - Preparing, 2-295, 2-297
- RPS
  - Enabling/disabling, 2-106
- RUN command, 2-300
  - Increased response time, 2-50

## S

- SAVE command, 2-304
- Saving \$STDLIST, 2-308
- Scheduled job
  - Abort, 2-15
- Scheduling jobs, 2-387
- SECURE command, 2-306
- Security
  - Altering file, 2-36
- Security in jobs, 2-179
- Security provisions
  - Removing from files, 2-266
  - Restoring, 2-306
- SEGMENTER command, 2-307
- Sending messages to sessions, 2-398
- Sending messages to the console, 2-400
- Sending messages to the operator, 2-400
- Sending urgent messages to jobs/sessions, 2-411
- Sequence numbers in job files, 1-6
- Serial Disks
  - formatting, initializing and labeling, 2-405
- Session
  - Abort, 2-14
  - Permit, 2-16
- Sessions

- Aborting, 2-217
  - creating on a device, 2-370
- Disabling, 2-265
- displaying status, 2-347
- Displaying status, 2-339
- Enabling, 2-218
- Ending, 2-75
- initiate, 1-1
- Input priority, 2-176
- Limiting number of, 2-181
- naming, 2-370
- sending messages, 2-398, 2-411
- spooling, 2-385
- Starting, 2-160
- Terminating, 2-75
- SETCATALOG command, 2-309
- SET command, 2-308
- SETDUMP command, 2-312
- SETJCW command, 2-314
- SETMSG command, 2-319
- Setting
  - Connect-time, 2-161
  - CPU-time, 2-161
- Setting the terminal speed, 2-360
- Sharing files, 2-132
- SHOWALLOCATE command, 2-320
- SHOWALLOW
  - ALLOW, 2-22
- SHOWALLOW command, 2-323
- SHOWCACHE command, 2-324
- SHOWCATALOG command, 2-326
- SHOWCOM command, 2-328
- SHOWDEV command, 2-330
- SHOWIN command, 2-333
- SHOWJCW command, 2-337
- SHOWJOB command, 2-339
- SHOWLOG command, 2-344
- SHOWLOGSTATUS command, 2-345
- SHOWME command, 2-347
- SHOWOUT command, 2-349
- SHOWQ command, 2-354
- SHOWTIME command, 2-356
- =SHUTDOWN command, 2-357
- SHUTQ command, 2-359
- Shutting down the system, 2-357
- Single disk caching, 2-369, 2-376
- Special
  - Capabilities, 2-1
- Special capabilities, 1-2
- Specifying terminal types, 2-164
- SPEED command, 2-360
- SPL/3000 programs
  - compiling, 2-362, 2-364, 2-366
  - executing, 2-364
  - preparing, 2-364, 2-366

- SPL command, 2-362
- SPLGO command, 2-364
- SPLPREP command, 2-366
- Spooled device
  - Clear, 2-12
- Spooled devices
  - suspending output, 2-393
- Spooler output
  - Resuming, 2-291
- Spoolfile characteristics
  - Changing, 2-40
- Spool files
  - Deleting from disk, 2-101
  - Output priority, 2-234
- Spooling
  - batch jobs, 2-385
  - jobs/sessions/data, 2-385
  - starting for a device, 2-374
  - stopping to a device, 2-377
- Spool queues
  - closing, 2-359
  - Opening, 2-233
- Stack dump facility
  - disabling, 2-279
  - initiation during ABORT, 2-312
- STARTCACHE command, 2-369
- Starting
  - spooling for a device, 2-374
- Starting batch jobs, 2-171
- Starting configuration dialog, 2-395
- Starting interactive sessions, 2-160
- Starting system shutdown, 2-357
- Starting user logging, 2-216
- START NORECOVERY
  - default settings, 2-402
- START RECOVERY
  - default settings, 2-402
- STARTSESS command, 2-370
- STARTSPOOL command, 2-374
- STD files, 2-131
- \$STDIN, 1-5, 2-123
- STDINX, 2-123
- \$STDLIST, 1-5, 2-123
  - Deleting, 2-308
  - Saving, 2-308
- STOPCACHE command, 2-376
- Stopping
  - spooling to a device, 2-377
- Stopping header/trailer output, 2-158
- Stopping user logging, 2-216
- STOPSPOOL command, 2-377
- STORE command, 2-379
- Storing files to tape, 2-379
- STREAM command, 2-385
- Streamed jobs
  - terminating, 2-387
- Streaming jobs, 2-387
- STREAMS command, 2-391
- Subqueue information
  - displaying, 2-354
- Subqueues, 2-402
- Subsystem commands
  - jobs, 2-388
- Subsystems
  - EDIT, 2-114
  - FCOPY, 2-121
- Suspended jobs
  - Resuming, 2-289
- Suspended operation
  - Resuming, 2-288
- Suspending
  - Jobs, 2-67
- Suspending output, 2-393
- SUSPENDSPOOL command, 2-393
- SWITCHLOG command, 2-394
- SYSDUMP command, 2-395
- System
  - Full backup, 2-153
- System backup, 2-395
  - partial, 2-236
- System console
  - Moving, 2-93
- System debugger
  - Entering, 2-100
- System-defined files, 1-5, 2-123
- System-defined JCWs, 2-316
- System library, 2-250
- System log file information
  - Displaying, 2-344
- System logging, 2-206
  - Resuming, 2-290
- System manager
  - definition, 1-2
- System-Reserved JCWs, 2-317
- System shutdown
  - starting, 2-357
- System supervisor
  - definition, 1-2

**T**

- TAKE command, 2-397
- Tape
  - storing files on, 2-379
- Tape density
  - Files, 2-128
- Tapes
  - Reading paper, 2-254
- TELL command, 2-398
- "Tell" messages
  - enabled or disabled, 2-319



- TELLOP command, 2-400
- Temporary files, 2-74
  - Displaying information, 2-195
  - Saved as permanent, 2-304
- Temporary Programs
  - Automatic Allocation, 2-50
- Terminals
  - baud rate, 2-361
- Terminal speed
  - setting, 2-360
- Terminal types
  - Specifying, 2-164
- Terminating
  - Sessions, 2-75
- Terminating a data set, 2-117
- Terminating IF blocks, 2-116
- Terminating processes, 2-279
- Terminating streamed jobs, 2-387
- Text files
  - Creating, 2-114
- Time, displaying, 2-356
- Trailer output
  - Resuming, 2-159
  - Stopping, 2-158
- TUNE command, 2-401
- TurboSTORE, 2-284

## U

- UDCs
  - activating, 2-309
  - Controlling execution, 2-169
  - Listing, 2-326
  - Overriding errors, 2-95
- UP command, 2-404
- Urgent messages, 2-411
- User
  - Attributes, 2-42
  - Capabilities, 2-22, 2-42
- User attributes, altering, 2-42
- User commands
  - Overriding errors, 2-95
- User control of devices
  - Removing, 2-105
- User-defined JCWs, 2-315
- User logging
  - displaying file status, 2-345
  - Restarting, 2-216
  - Starting, 2-216
  - Stopping, 2-216
- User logging identifiers
  - Changing, 2-34

- Removing, 2-269
- User passwords, 2-228
- Users
  - Allowing operator control of a device, 2-48
  - defining new, 2-228
  - Deleting, 2-260
  - displaying for volume sets, 2-410
  - Displaying information, 2-208
- Users' logging identifiers, 2-154
- Using passwords, 2-163
- Using the command interpreter, 2-162
- USL file
  - COBOL, 2-80
  - COBOLII, 2-86
- USL files, 2-143, 2-252
  - with PREP command, 2-246
- Utilities
  - ASOCTBL, 2-48

## V

- VINIT command, 2-405
  - functions, 2-405
- VMOUNT command, 2-408
- Volumes
  - Foreign disk, 2-138
- Volume sets
  - Cancelling, 2-180
  - defining new, 2-231
  - Deleting, 2-261
  - displaying users, 2-410
  - Groups, 2-226
  - Listing definitions, 2-211
  - Modifying the definition, 2-46
  - naming, 2-408
  - Releasing, 2-107
  - reserving, 2-219
  - Reserving, 2-214
- VSUSER command, 2-410

## W

- Waiting job
  - Abort, 2-15
- WARN command, 2-411
- "Warn" messages
  - enabled, 2-319
- WELCOME command, 2-413
- Welcome messages
  - creating, 2-413

## X

- X-OFF control character, 2-254

**Part No. 32033-90006**  
**Printed in U.S.A. 1989**

E1089

