

decsystem10

Operating System Command Manual

digital



**dec**system10  
**OPERATING SYSTEM  
COMMANDS MANUAL**

DEC-10-OSCMA-A-D

This manual reflects the software of the 5.07 and  
6.01 releases of the monitor.

*The DECsystem-10 Commands Manual* was produced via RUNOFF (version 10) and converted to Typeset-10. Software enhancements will enable us to attain even better formatting and typographical quality in the near future.

This document reflects the software associated with the 5.07 and 6.01 releases of the Monitor. For individual system program numbers, refer to page v.

First Printing, May 1971  
Revision: December 1971  
Revision: June 1972  
Revision: April 1973  
Revision: May 1974

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

DON  
8901295  
ECOM

The software described in this document is furnished to the purchaser under a license for use on a single computer system and can be copied (with inclusion of DIGITAL's copyright notice) only for use in such system, except as may otherwise be provided in writing by DIGITAL.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

*Copyright ©1971, 1972, 1973, 1974,  
by Digital Equipment Corporation*

The postage prepaid READER's COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation.

CDP	DIGITAL	INDAC	PS/8
COMPUTER LAB	DNC	KA10	QUICKPOINT
COMSYST	EDGRIN	LAB-8	RAD-8
COMTEX	EDUSYSTEM	LAB-8/e	RSTS
DDT	FLIP CHIP	LAB-K	RSX
DEC	FOCAL	OMNIBUS	RTM
DECCOMM	GLC-8	OS/8	RT-11
DECTAPE	IDAC	PDP	SABR
DIBOL	IDACS	PHA	TYPESET-10
			UNIBUS



# FOREWORD

*DECsystem-10 Operating System Commands* is a complete reference document describing the commands available in the DECsystem-10 operating system. The information presented in this manual reflects the 6.01/5.07 release of the monitor and other related programs. Commands to both the monitor command language interpreter and the programs in the Batch system are grouped in alphabetical order for easy reference to the command repertoire.

*DECsystem-10 Operating System Commands* does not include reference material on assembly language programming. This information can be found in *DECsystem-10 Monitor Calls* (DEC-10-MRRD-D), which is intended for the experienced assembly language programmer. Included in *DECsystem-10 Monitor Calls* are discussions of the monitor programmed operators and the various I/O devices connected to the system. The two manuals, *DECsystem-10 Operating System Commands* and *DECsystem-10 Monitor Calls*, supersede the *Timesharing Monitors Programmer's Reference Manual* (DEC-T9-MTzd-D) and all of its updates.

A third manual, *Introduction to DECsystem-10 Software* (DEC-10-MZDB-D), is a general overview of the DECsystem-10. It is written for the person, not necessarily a programmer, who knows computers and computing concepts and who desires to know the relationship between the various components of the DECsystem-10. This manual is not intended to be a programmer's reference manual, and therefore, it is recommended that it be read once before reading the above-mentioned reference documents.

## SYNOPSIS OF DECsystem-10 OPERATING SYSTEM COMMANDS

Chapter 1 presents all of the commands available to the user and introduces the various components of the operating system that interface with the user. Chapter 2 is a detailed description of the commands processed by the monitor command language interpreter. Presented in Chapter 3 are the commands to the Batch system and a discussion of the programs in this system. The DECsystem-10 system error messages and error codes are listed in Chapter 4 along with descriptive information on how to correct the errors. The appendices contain supplementary reference material and tables.

## CONVENTIONS USED IN DECsystem-10 OPERATING SYSTEM COMMANDS

The following conventions have been used throughout this manual:

- |      |   |
|------|---|
| dev: | Any logical or physical device name. The colon must be included when a device is used as part of a file specification.  |
| list | A single file specification or a string of file specifications. A file specification consists of a filename (with or without a filename extension), a device name if the file is not on disk, a project-programmer number, if the file is not in the user's disk area, and a protection code. |

arg	A pair of file specifications or a string of pairs of file specifications.
jobn	Any job number assigned by the monitor.
file.ext	Any legal filename and filename extension.
core	Decimal number of 1K blocks of core.
adr	An octal address.
C(adr)	The contents of an octal address.
[proj,prog]	Project-programmer numbers; the square brackets must be included in the command string.
fs	Any legal file structure name or abbreviation.
\$	The symbol used to indicate an altmode.
^x	A control character obtained by depressing the CTRL key and then the character key x.
←	A back arrow used in command strings to separate the input and output file specifications.
*	The system program response to a command string.
.	The monitor response to a command string.
)	The symbol used to indicate that the user should depress the RETURN key. This key must be used to terminate every command to the Monitor Command Language Interpreter.
—	Underscoring used to indicate computer typeout.
n	A decimal number.
=	An equal sign used in command strings to separate the input and output file specifications.

### SOFTWARE VERSION NUMBERS

ALCFIL	Version 7	KJOB	Version 50
BATCON	Version 10	LINED	Version 13A
COMPIL	Version 22A	LINK-10	Version 1A
COPY	Version 6A	LOGIN	Version 54
CREF	Version 47	Monitor	6.01/5.07
DAEMON	Version 7	OMOUNT	Version 23
DIRECT	Version 3	OPSER	Version 5
DUMP	Version 4	PIP	Version 33
FAILSA	Version 77	PLEASE	Version 11
FILCOM	Version 20A	QUEUE	Version 5
FILEX	Version 15	QUOLST	Version 4
FUDGE2	Version 15	REATA	Version 3
GLOB	Version 5A	SETSRC	Version 12
GRIPE	Version 3	SPRINT-10	Version 1
HELP	Version 4	SYSTAT	Version 471
INITIA	Version 3	TECO	Version 23
		UMOUNT	Version 21



# CONTENTS

	Page
<b>CHAPTER 1 INTRODUCTION</b>	
1.1 Jobs .....	1-1
1.2 Monitor Mode and User Mode.....	1-2
1.3 Command Interpreters .....	1-2
1.3.1 Monitor Command Language Interpreter .....	1-3
1.3.1.1 Special Characters .....	1-4
1.3.2 Batch Command Interpreter .....	1-5
1.4 Command Formats .....	1-6
1.4.1 Command Names .....	1-6
1.4.2 Command Arguments .....	1-7
1.4.2.1 Project-Programmer Numbers .....	1-7
1.4.2.2 Device Names.....	1-7
1.4.2.3 File Structure Names .....	1-10
1.4.2.4 File Specifications .....	1-10
1.4.2.5 Date-Time Arguments.....	1-12
1.4.3 Option Files .....	1-14
1.5 COMPIL-Class Commands .....	1-15
1.5.1 Indirect Commands(@ Construction).....	1-17
1.5.2 The + Construction .....	1-17
1.5.3 The = Construction .....	1-17
1.5.4 The < > Construction .....	1-18
1.5.5 Compile Switches.....	1-18
1.5.6 Standard Processor .....	1-19
1.5.7 Processor Switches.....	1-19
1.5.8 Switches to the Linking-Loaders.....	1-24
1.5.8.1 Passing LOADER Switches .....	1-24
1.5.8.2 Passing LINK-10 Switches .....	1-25
<b>CHAPTER 2 SYSTEMS COMMANDS AND PROGRAMS</b>	
2.1 Commands by Functional Group.....	2-1
2.1.1 Job Initialization Commands.....	2-1
2.1.2 Facility Allocation Commands .....	2-1
2.1.3 Source File Preparation Commands.....	2-3
2.1.4 File Manipulation Commands.....	2-3
2.1.5 Object Program Preparation Commands.....	2-4
2.1.6 Object Program Control Commands .....	2-5
2.1.7 Object Program Examination Commands .....	2-6
2.1.8 Multiple Job Control Commands .....	2-7
2.1.9 Job Termination Commands .....	2-7
2.1.10 Sending Messages .....	2-7
2.1.11 Job Information Commands .....	2-8
2.1.12 System Information Commands .....	2-8
ALCFIL Program .....	2-9
ASSIGN Command.....	2-11
ATTACH Command.....	2-13

	Page
BACKSPACE Command .....	2-15
CLOSE Command .....	2-17
COMPILE Command .....	2-19
CONTINUE Command .....	2-26
COPY Command .....	2-27
CORE Command .....	2-29
CPUNCH Command .....	2-31
CREATE Command .....	2-37
CREF Command .....	2-38
CSTART Command .....	2-40
CCONTINUE Command .....	2-40
D(eposit) Command .....	2-43
DAYTIME Command .....	2-45
DCORE Command .....	2-46
DDT Command .....	2-52
DEASSIGN Command .....	2-54
DELETE Command .....	2-61
DETACH Command .....	2-63
DIRECT Command .....	2-64
DISMOUNT Command .....	2-72
DSK Command .....	2-74
DTCOPY Command .....	2-76
DUMP Command .....	2-79
DUMP Program .....	2-81
E(xamine) Command .....	2-89
EDIT Command .....	2-90
EOF Command .....	2-91
EXECUTE Command .....	2-92
FAILSAFE Command .....	2-99
FILCOM Program .....	2-106
FILE Command .....	2-115
FILEX Command .....	2-119
FINISH Command .....	2-122
FUDGE Command .....	2-124
FUDGE2 Command .....	2-126
GLOB Program .....	2-134
GRIPE Program .....	2-138
HALT Program .....	2-139
HELP Program .....	2-140
INITIA Command .....	2-144
JCONTINUE Command .....	2-145
KJOB Command .....	2-146
LABEL Command .....	2-151
LIST Command .....	2-153
LOAD Command .....	2-155
LOCATE Command .....	2-161
LOGIN Command .....	2-162
MAKE Command .....	2-164
MOUNT Command .....	2-166
OPSER Program .....	2-170
PJOB Command .....	2-174
PLEASE Command .....	2-175

	Page
PLOT Command .....	2-177
PRESERVE Command.....	2-183
PRINT Command .....	2-184
PROTECT Command.....	2-191
QUEUE Command .....	2-193
QUOLST Program.....	2-210
R Command .....	2-212
REASSIGN Command.....	2-214
REATTA Program.....	2-216
REENTER Command .....	2-218
RENAME Command .....	2-219
RESOURCES Command .....	2-221
REWIND Command.....	2-222
RUN Command.....	2-223
SAVE Command .....	2-225
SCHED Command .....	2-227
SEND Command.....	2-228
SET BLOCKSIZE Command .....	2-230
SET BREAK Command.....	2-231
SET CDR Command .....	2-234
SET CPU Command.....	2-235
SET DENSITY Command.....	2-237
SET DSKFUL Command .....	2-238
SET DSKPRI Command .....	2-239
SET HPQ Command .....	2-240
SET PHYSICAL LIMIT Command.....	2-242
SET SPOOL Command .....	2-244
SETSRC Command.....	2-246
SET TIME Command.....	2-250
SET TTY Command.....	2-252
SET VIRTUAL LIMIT Command.....	2-258
SET WATCH Command .....	2-259
SKIP Command.....	2-262
SSAVE Command .....	2-264
START Command.....	2-266
SUBMIT Command .....	2-267
SYSTAT Command .....	2-273
TECO Command.....	2-280
TIME Command .....	2-282
TPUNCH Command.....	2-284
TYPE Command .....	2-290
UNLOAD Command .....	2-291
USESTAT Command .....	2-292
VERSION Command .....	2-293
WHERE Command.....	2-296
ZERO Command.....	2-297

## CHAPTER 3 BATCH SYSTEM COMMANDS

3.1	Batch Commands.....	3-1
3.1.1	SPRINT-10 .....	3-1

	Page
3.1.2	The Queue Manager ..... 3-2
3.1.3	The Batch Controller ..... 3-2
3.1.4	The Output Spoolers ..... 3-4
3.2	Submitting Jobs ..... 3-4
3.2.1	Submitting a Job with Cards ..... 3-5
3.2.1.1	The \$JOB Card ..... 3-5
3.2.1.2	The \$PASSWORD Card ..... 3-5
3.2.1.3	The \$FORTRAN Card ..... 3-5
3.2.1.4	The \$EXECUTE Card ..... 3-6
3.2.1.5	The \$EOJ Card ..... 3-6
3.2.1.6	Output ..... 3-6
3.2.2	Submitting a Job with a File ..... 3-6
3.2.2.1	Image of the \$JOB Card ..... 3-6
3.2.2.2	Image of the \$FORTRAN Card ..... 3-7
3.2.2.3	Image of the \$EXECUTE Card ..... 3-7
3.2.2.4	Running SPRINT-10 ..... 3-7
3.2.3	Submitting a Job with a Control File to the Batch Controller ..... 3-8
3.2.4	Interjob Dependency ..... 3-8
3.3	SPRINT-10 Control Cards ..... 3-9
3.3.1	\$ALGOL Card ..... 3-10
3.3.2	\$BLISS Card ..... 3-12
3.3.3	\$COBOL Card ..... 3-14
3.3.4	\$DATA Card ..... 3-16
3.3.5	\$DECK Card ..... 3-20
3.3.6	\$DUMP Card ..... 3-23
3.3.7	\$EOD Card ..... 3-23
3.3.8	\$EOJ Card ..... 3-24
3.3.9	\$ERROR and \$NOERROR Card ..... 3-24
3.3.10	\$EXECUTE Card ..... 3-25
3.3.11	\$FORTRAN and \$F40 Cards ..... 3-25
3.3.12	\$INCLUDE Card ..... 3-27
3.3.13	\$JOB Card ..... 3-28
3.3.14	\$MACRO Card ..... 3-32
3.3.15	\$MODE Card ..... 3-34
3.3.16	\$PASSWORD Card ..... 3-36
3.3.17	\$RELOCATABLE Card ..... 3-36
3.3.18	\$SEQUENCE Card ..... 3-37
3.3.19	\$SNOBOL Card ..... 3-37
3.4	Batch Control File Commands ..... 3-39
3.4.1	.BACKTO Command ..... 3-41
3.4.2	.CHKPNT Command ..... 3-41
3.4.3	.ERROR Command ..... 3-42
3.4.4	.GOTO Command ..... 3-42
3.4.5	.IF Command ..... 3-43
3.4.6	.NOERROR Command ..... 3-43
3.4.7	.NOOPERATOR Command ..... 3-44
3.4.8	.OPERATOR Command ..... 3-44
3.4.9	.REQUEUE Command ..... 3-45
3.4.10	.REVIVE Command ..... 3-46
3.4.11	.SILENCE Command ..... 3-46
3.5	Job Output ..... 3-46
3.5.1	The Log File ..... 3-47



		Page
3.5.1.1	SPRINT-10 Messages .....	3-47
3.5.1.2	SPRINT-10 Error Reporting .....	3-48
3.5.1.3	Batch Controller Messages .....	3-48
3.5.1.4	Batch Controller Error Reporting .....	3-49
3.6	Sample Jobs .....	3-49

**CHAPTER 4    SYSTEM DIAGNOSTIC MESSAGE AND ERROR CODES**

APPENDIX A	STANDARD SYSTEM NAMES
APPENDIX B	CARD CODES
APPENDIX C	TEMPORARY FILES
APPENDIX D	SAVE AND SSAVE COMMANDS
APPENDIX E	SWITCHES
INDEX	



# CHAPTER 1

## INTRODUCTION

The DECsystem-10 Operating System is the interface between the user and the actual machine. The operating system, or monitor, has many functions, some of which are:

1. scheduling multiple and simultaneous use of the system,
2. protecting users of the system from one another,
3. allowing access to system resources including peripheral devices,
4. providing a comprehensive disk file system,
5. directing data flow between peripheral devices and the user's program,
6. controlling non-interactive jobs, and
7. overlapping input-output operations with computations for high system efficiency.

The user communicates with the operating system by means of the monitor command language. With the command language he may access all available resources of the computing system and obtain all the services provided by the operating system.

### 1.1 JOBS

The DECsystem-10 computing system is a multiprogramming system; that is, control is transferred rapidly among a number of jobs in such a way that all jobs appear to be running simultaneously. The term job refers to the entire sequence of steps, from beginning to end, that the user initiates from his interactive terminal or card deck or that the operator initiates from his operator's console. When a user initiates a job from his interactive terminal, the beginning of the job is designated by the LOGIN command and the end by the KJOB command. If a user initiates a job with a card deck, the beginning of the job is the \$JOB card and the end is the \$EOJ or the end-of-file card. Operator jobs usually begin when the system is initialized and end when the system goes down.

Jobs, which may be timesharing, batch, or real-time in nature, may be initiated at the central computer site or at remote locations connected by the telephone system. Once a user initiates a job, it is possible for him to initiate another job without killing the first one. For example, a user can initiate a timesharing job and by using the SUBMIT monitor command submit a second job for batch processing (refer to Chapter 2). He may then wait for the results from this batch job, or have the results automatically output while he continues his timesharing job.

In configuring and loading the DECsystem-10, the system administrator sets the maximum number of jobs that his system can simultaneously handle. This number may be up to 127

jobs if the system has enough memory, disk storage, processor capacity, and terminals to handle this load.

## 1.2 MONITOR MODE AND USER MODE

From the timesharing user's point of view, his terminal is in either monitor mode or user mode. In monitor mode, each line the user types in is sent to the monitor command language interpreter. The execution of certain commands (as noted in the following examples) places the terminal in user mode. When the terminal is in user mode, it becomes simply an I/O device for that user. In addition, user programs use the terminal for two purposes. The user program will either accept user command strings from the terminal (user mode) or use the terminal as a direct I/O device (data mode).

Example (terminal dialogue):

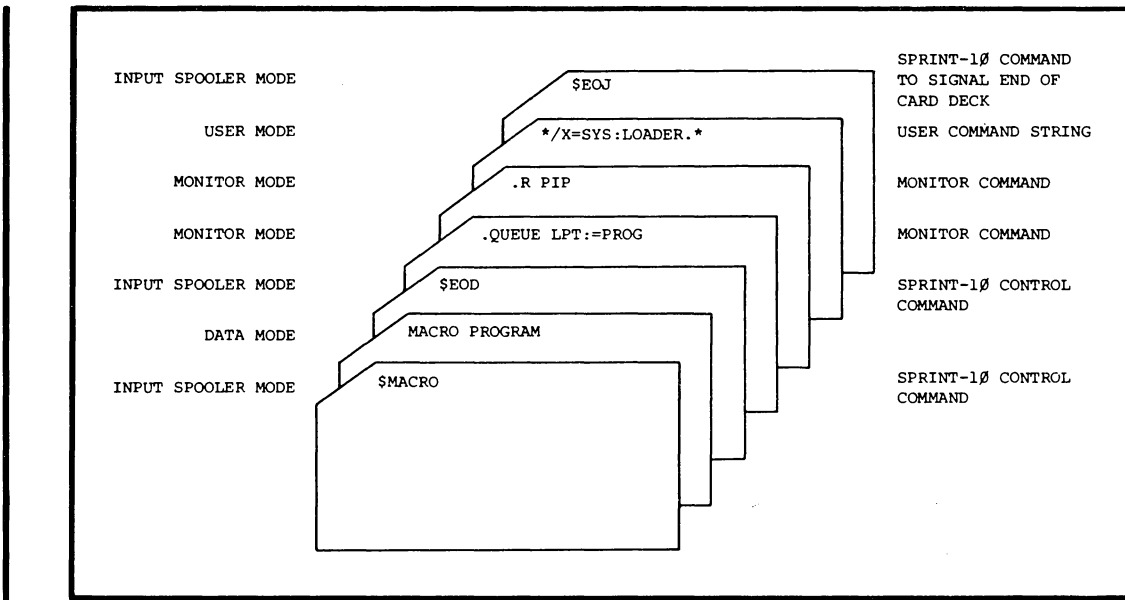
monitor mode	<code>.R PIP</code>	monitor command
user mode	<code>*DSK:PROG1.MAC TTY:</code>	user command string
data mode	<code>THIS IS FILE 1 ^Z</code>	user program using terminal as input device
	<code>*^C</code>	
monitor mode	<code>.R MACRO</code>	monitor command
user mode	<code>*.TTY: DSK:PROG1</code>	user command string
data mode	<code>:</code>	user program using terminal as an output device
	<code>:</code>	
	assembly listing	
	<code>:</code>	
	<code>:</code>	

The special character `^C` (produced by typing `C` with the `CONTROL` key depressed) is used by a timesharing user to stop a user program and return the terminal to monitor mode. If the user program is waiting for input from the terminal, the user needs to type only one `^C` to return the terminal to monitor mode; otherwise, he must type two `^C`'s. Because of this procedure, the user knows that his program is not waiting for input if there is not response from the monitor after one `^C`. Certain commands cause the user program to start running or to continue (as noted in the following chapter) but leave the terminal in monitor mode.

When the system is started, each terminal is in monitor mode ready for users to log in. However, if the system becomes fully loaded (i.e., the maximum number of jobs that the system is set to handle has been initiated), then any unused terminals from which access is requested will receive the message `JOB CAPACITY EXCEEDED`.

The card-oriented Batch user can think of his cards as being in input spooler mode, monitor mode, or user mode. When the card is in input spooler mode, it contains a control command beginning with a `$` (refer to Chapter 3) and is sent to the Spooler, `SPRINT`. `SPRINT` interprets these commands and performs various actions to create a control file for the Batch Controller. When the card is in monitor mode, it contains a monitor command preceded by a period and is copied by `SPRINT` into the control file. When the card is in user mode, it contains a user-level program command preceded by an asterisk or an equal sign and is also copied by `SPRINT` into the control file. As each line in the control file is executed, the Batch Controller passes the monitor-level line to the monitor command language interpreter and the user-level line to the user program.

Example (sample card deck):



## 1.3 COMMAND INTERPRETERS

### 1.3.1 Monitor Command Language Interpreter

When the terminal is in monitor mode, the user communicates with the monitor command language interpreter. By means of commands to this interpreter, the user may initialize jobs, allocate facilities, prepare source files, manipulate files, prepare, control, and examine object programs, control job sequences and multiple jobs, terminate jobs, send messages, and obtain job and system information. The commands described in Chapter 2 are processed by this interpreter.

Most commands are processed without delay. However, a command may be momentarily delayed if a job is swapped out to the disk and the command requires that the job be resident in core; the command is executed when the job is swapped into core. The completion of each command is signaled by the output of a carriage return, line feed sequence. If the terminal is left in monitor mode, a period follows the carriage return, line feed. If the terminal is left in user mode, any response other than a carriage return, line feed comes from the user's program. For example, most standard system programs immediately send an asterisk(\*) to the user's terminal to indicate their readiness to accept user command strings.

The type-ahead technique may be employed by the experienced timesharing user at a terminal. This means that the user does not have to wait for the completion of one command before he can begin another. For example, if two operations are desired from the monitor, the

request for the second operation can be typed before receiving the period after completion of the first.

The command interpreter makes several checks before processing commands from users. On disk systems, if a user who has not logged in types a command that requires him to be logged in, the system responds with

?LOGIN PLEASE

and the user's command is not executed. The commands discussed in Chapter 2 all require login except where explicitly stated otherwise. When a command is recognized that requires the job to have core and the job has no core allocated, the command interpreter responds with

?NO CORE ASSIGNED

and the user's command is not executed.

**1.3.1.1 Special Characters** - There are several special characters recognized by the monitor command language interpreter that causes specific functions to be performed. As noted previously, control-C (^C) interrupts the program that is currently running and returns the terminal to monitor mode. This character causes the input line back to the last break character (e.g., carriage return, line feed) to be deleted (equivalent to the action of a ^U). Two control-^C's are necessary if the user program is not requesting input from the terminal (i.e., the program is in the middle of execution).

The RUBOUT key on the terminal generates a character that causes the last character typed to be deleted. This permits correction of typing errors. Depressing the RUBOUT key n times causes the last n characters typed to be deleted. The deleted characters are echoed on the terminal enclosed in backslashes (\). Characters beyond the last break character or characters already processed by the user program are not deleted.

Control-U (^U) causes the deletion of the current input line, back to the last break character. The system responds with a carriage return, line feed so that the line may be typed again. Once a break character has been typed, line-editing features (^U and RUBOUT) can no longer be used on that line, except when running TECO.

Control-O (^O) temporarily suppresses output to the terminal. This action is useful when a program begins output of a long message which does not interest the user. If he does not want to wait for his terminal to finish printing the message, he can stop the output in one of two ways. He can type two control-C's but this action will also stop execution of the program. Alternatively, the user can type ^O and the program continues to execute but its output is not printed on the terminal. The system responds with a carriage return, line feed sequence. Output is reinstated to the terminal when one of the following conditions occur:

1. The executing program requests input from the terminal.
2. The program terminates and returns control to the monitor.
3. The user types ^C to return to the monitor.
4. The user types another ^O.

At remote stations, the effect of the ^O may be somewhat delayed.

Control - T (^T) - causes the terminal to print status information pertaining to the current user job. The status information returned is the same as the information returned for the USESTAT command, namely:

1. incremental day time (i.e., time since user last issued a ^T or USESTAT command.)
2. incremental run time (i.e., CPU time used since last ^T, USESTAT or TIME.)
3. incremental disk reads (i.e., disk blocks read since last use of ^T, USESTAT or DSK.)
4. incremental disk writes (i.e., disk blocks written since last use of ^T, USESTAT or DSK.)
5. program name
6. core size
7. job state
8. program counter (i.e., the virtual address of the next instruction that the program will execute.)

Control-T is provided so that a user at a terminal can determine the progress of his job without interrupting the execution of the job. Thus, a user whose terminal has not typed anything for a minute or so can type a ^T to determine the state of his job without interfering with the job's execution. Note that the Control-T character itself is not passed to the job as an input character. However, some programs activate a special interrupt feature (see Paragraph 3.1.4 in the *Monitor Calls Manual*). When a ^T is typed, control of such programs automatically transfers to a routine used to output status information, in addition to the regular USESTAT printout. (For example, a text editor in the midst of a search might print out information about the progress of the search.)

Control-R (^R) retypes the current input line after all rubout processing. For example, if a user types in a line incorrectly, then makes corrections using the RUBOUT key, the corrected line may be retyped in its entirety by typing ^R. An example of this is:

```
SET TTQ\Q\Y N\O\O FILE\EL ^R
SET TTY NO FILL
```

Control-R will issue a carriage return/line feed before printing the corrected input line.

### 1.3.2 Batch Command Interpreter

The monitor command language interpreter is used for all monitor commands submitted via the Batch system. In addition, the Batch user issues commands that are only used by the Batch programs: the input spooler (SPRINT) and the Batch Controller (BATCON). Control

commands, discussed in Chapter 3, are processed by the input spooler and, by means of these commands, the user can create a control file, a log file, and data files; can enter jobs into the Batch input queue; and can insert monitor commands into the control file. An additional interpretation is done by the Batch Controller. When the job is executed, the Batch Controller processes the control file to pass monitor commands to the monitor command language interpreter and user-level commands to the appropriate programs.

## 1.4 COMMAND FORMATS

Each command is a line of ASCII characters in upper and/or lower case. Spaces and TABs preceding the command name are ignored. Comments may be typed on the same line as the command by preceding the comment with a semicolon. The monitor and batch command language interpreters do not interpret or execute a line of comments. Every command line to the monitor command interpreter should be terminated by pressing the RETURN key on the console. In examples in this manual, the symbol ↵ is used to indicate that the user should depress the RETURN key. If the command line is in error, all of the line preceding the error is typed out by the monitor preceded and followed by a ?, and the terminal remains in monitor mode.

The user can continue command lines to several system programs (e.g., DIRECT, DUMP, QUEUE, and LINK-10) by placing a hyphen as the last nonblank, noncomment character before the carriage return-linefeed (i.e., before pressing the RETURN key). Continuation lines are treated as part of the current command line, which is not considered terminated until a carriage return-linefeed is seen without a preceding hyphen. This allows command lines to the above-mentioned programs to be indefinitely long.

### 1.4.1 Command Names

Commands to the monitor command interpreter are alphabetic strings of one to six characters; characters after the sixth are ignored. Only enough characters to uniquely identify the command need be typed. It is recommended that a Batch job use the full command name since the number of characters required may change when new commands are added to the monitor. Thus, a card deck or control file with abbreviated commands may no longer function properly.

Installations choosing to implement additional commands are advised to preserve the uniqueness of the first three letters of existing commands.

Control commands to the Stacker in the multiprogramming batch system must have a dollar sign (\$) in the first column of the card or the line and an alphabetic character in the second column. Only the first part of the command name need be specified; as long as the specified command name is unique, it is accepted. The first three characters of the command name are generally sufficient to ensure uniqueness.



## 1.4.2 Command Arguments

Arguments follow the command name and are separated from it by a space or TAB. If the monitor command interpreter recognizes a command name, but a necessary argument is missing, the monitor responds with

?TOO FEW ARGUMENTS

Extra arguments are ignored.

**1.4.2.1 Project-Programmer Numbers and Passwords** – Access to the DECsystem-10 is limited to authorized users. The system administrator provides each authorized user with a project number, a programmer number, and a password. The project numbers range from 1 to 377777 octal (numbers 1 to 10 are reserved for DEC) and the programmer numbers range from 1 to 777777 octal (numbers 1 to 7 are reserved for DEC and numbers 400000 to 777777 are reserved for special purposes)(1). These numbers identify the user and his file storage area on a file structure. In a command string, the project and programmer numbers are separated with a comma and must be enclosed in square brackets, e.g., [10,7].

The password is from one to SIXBIT characters and is only used when logging on the computing system. To maintain password security, the monitor does not echo the password. On terminals with local copy (refer to DECsystem-10 Monitor Calls), a mask is typed to make the password unreadable.

**1.4.2.2 Device Names** – Associated with each system device controlled by the computing system is a physical device name. This name consists of three letters, zero to three numerals specifying the unit number, and a colon. Table 1-1 lists the generic physical device names associated with the various system devices. Refer to Appendix A for the list of special disk devices that are predefined in the DECsystem-10.

The user may also associate a logical device name with a particular physical device. The logical name is from one to six alphanumeric characters of the user's choice, followed by a colon, and can be used synonymously with the physical device name in any reference to the device. Logical device names allow the user, when writing his program, to use arbitrarily selected device names, which he assigns to the most convenient physical devices at run time. However, care should be exercised when assigning logical device names because these names have priority over physical device names. For example, if a DECTape is assigned the logical name DSK, then all of the user's programs attempting to use the disk via the device name DSK will use the DECTape instead.

Except for disk devices, only one logical device name can be associated at any one time with a physical device. The same logical name can be used for a second physical device by disassociating it from the first device and associating it with the second device via the ASSIGN command. Logical device names are disassociated from devices with the DEASSIGN command (refer to Chapter 2). Subsequent ASSIGN commands (refer to Chapter 2) to devices except disk devices replace the old logical name with the new one.

---

1

When the programmer number is from 1 to 7, all project numbers are reserved for DEC.

Table 1-1  
System Devices

Device	Generic Physical Device Name
Card Punch	CDP:
Card Reader	CDR:
Console TTY	CTY:
DECtape	DTx: +
Disk	DSK:
Packs	DPx: +
Fixed-Head	FHx: +
Display	DIS:
Line Printer	LPT:
Magnetic Tape	MTA:
Operator Terminal	OPR:
Paper-tape Punch	PTP:
Paper-tape Reader	PTR:
Plotter	PLT:
Pseudo-TTY	PTY:
System Library	SYS:
Terminal	TTY:

X represents A,B,...., indicating the first controller, second controller, etc.

The following is an example of the use of physical and logical device names.

.ASSIGN DTA: ABC:	User requests a DECtape drive with the logical name ABC.
DEVICE DTA6 ASSIGNED	Monitor has given the user drive DTA6. The user mounts a DECtape on drive DTA6.
.ASSIGN PTP: ABC:	User requests the paper-tape punch with the logical name ABC.
% LOGICAL NAME WAS IN USE. PTP ASSIGNED	Paper-tape punch is reserved, and ABC now refers to the PTP.
.R PIP	User requests the system program PIP (Peripheral Interchange program).
*ABC: DTA6:FILEA	User issues a command string to PIP asking that file FILEA be transferred from device DTA6 to logical device ABC (physical device PTP: which is assigned to the user).

(continued on next page)

*^C	User returns to the monitor mode.
.ASSIGN DTA: DEF:	User requests another DECtape drive with logical name DEF.
.ASSIGNED TO JOB N1,N2,...	All drives are in use by the specified jobs. No DECtape drive is assigned, and no logical assignment is made.
.ASSIGN DTA6: DEF:	User requests drive DTA6 (which he already has) with logical name DEF. The copy of the directory currently in core is cleared.
DEVICE DTA6 ASSIGNED	User mounts a new DECtape on the previously assigned drive. The new DECtape directory is read into core when next accessed.
.DEASSIGN PTP:	User deassigns PTP, thereby clearing the logical name ABC.
.R PIP	User requests PIP.
*ABC: DEF:FILEB	User requests that file FILEB be transferred from device DEF to device ABC.
?DEVICE ABC DOES NOT EXIST	The logical device name ABC is no longer assigned.
*^C	User returns to monitor mode.
.ASSIGN DTA6: XYZ:	User requests drive DTA6 again with logical name XYZ. The logical name DEF is no longer associated with DTA6. The old directory is cleared from core.
DEVICE DTA6 ASSIGNED	User mounts a new DECtape. The new directory is read into core when accessed.

**1.4.2.3 File Structure Names** – Disk devices are grouped according to file structures, which are logical arrangements of 128-word blocks on one or more disk units of the same type. Examples of types of disk units are: an RP02 disk pack or an RM10B drum. Although a file structure can exist on exactly one disk unit, it can be distributed over several disk units of the same type and designated by a single name. However, two file structures cannot exist on the same unit. Each file structure has a SIXBIT name specified by the operator at structure definition time. This name can consist of five or less alphanumeric characters and must not duplicate a physical device name, a unit name, or an existing file structure name. The recommended names for public file structures are DSKA, DSKB,..., DSKN in order of decreasing speed.

**1.4.2.4 File Specifications** – All information (programs and data) in the system is stored as named files. Each named file has associated with it a file specification which consists of

1. the physical device name or file structure name,
2. the filename,
3. the filename extension,
4. the ordered list of directory names, and
5. the access protection code.

The first four items of the file specification are necessary to uniquely identify a disk file. File specifications are ignored when given for devices other than DECTape or disk since these two devices are the only directory-oriented devices. In addition, items 4 and 5 do not apply to DECTapes.

The physical device name used for DECTape or the file structure name used for disk may be any legal device name discussed in the foregoing sections. A colon should always follow the device name; e.g., DTA3:. The filename is from one to six SIXBIT characters; all characters after the sixth are ignored. The filename extension is a period followed by zero to three characters and is used to indicate the type of information in the file. (Refer to Appendix A for a list of standard filename extensions.) It is recommended that only the standard extensions be used even though other extensions are valid. Most programs only recognize filenames and extensions consisting of letters and digits. The ordered list of directory names identifies the disk area in which the file is stored. This list can be a user file directory (UFD) represented by the project-programmer number of the owner of the files in the directory or can be a user file directory followed by one or more sub-file directories (SFDs). (Refer to the DECsystem-10 Monitor Calls for a description of SFDs.) The directory name must be enclosed in square brackets. The access protection of the file is a three-digit code designating which users can read or write the file and must be enclosed in angle brackets. The protection code is specified only for output files. For a given file, the users are divided into three groups: the owner of the file, the users with the same project number as the owner, and the rest of the users. The standard protection code is 057 which allows users in the owner's project to read and execute the file and prevents access by all other users. (For a complete description of access protection, refer to DECsystem-10 Monitor Calls.) The standard protection code can be redefined by the various installations.

In command strings, the filename, the device name if the file is not on disk, and the directory name if the file is not in the user's disk area, are required. The filename extension, the device name if the file is on the disk, the directory name if the file is in the user's disk area, and the protection code are optional. The following are examples of file specifications:

TEXT.MAC	filename and extension
DTA3:FILEA	device and filename
DSK:PROG2.CBL[10,16]	device, filename, extension, and directory name
DSKA:MAIN.F4[27,235]057>	complete file specification

Many command strings allow the wildcard construction to be used. This means that the filename, the extension, or the directory name may be replaced totally with an asterisk or partially with a question mark to designate certain filenames, extensions, or directories. The asterisk is used as a wild field to designate the entire filename, extension, or directory name. For example,

filename.*	All files with this filename and any extension.
*.ext	All files with this extension and any filename.
*.*	All files.
*.*[project,*]	All files in directories with this project number and any programmer number.

The question mark is used as a wild character to designate part of the filename, extension, or directory name. A question mark is used for each character that is to be matched; i.e., PR?? matches on four characters or less. For example,

filename .M??	All files with this filename and any extension beginning with M.
TES?? .ext	All files with this extension and any filename up to 5 characters beginning with TES.
?? .???	All files with filenames of two characters or less.
[25,5??]	All files in directories with the project number 25 and the programmer numbers 500-577.

The asterisk and the question mark can be specified together in the same construction.

??.*	All files with filenames of two characters or less.
------	---

In addition, the directory name can be specified with the project number, the programmer number, or both numbers missing. The following are examples of the various ways of representing a particular directory.

[15,23]	The UFD [15,23].
[,30]	The UFD that has the user's project number and the specified programmer number (30).
[36,]	The UFD that has the specified project number (36) and the user's programmer number.
[.]	The user's UFD.
[-]	The user's default directory which may be different from his UFD (refer to the DECsystem-10 Monitor Calls and the SETSRC program).
[.,SUB1,SUB2]	The sub-file directory SUB2 under the sub-file directory SUB1 in the user's UFD.

The number sign can be used to represent a filename or extension that contains characters that cannot be typed because they have special meanings in the system. For example, if a file with the name \*.MAC were typed in a command string, the user would be referencing all files with the extension .MAC since the \* designates all files with the specified extensions MAC. To allow a filename or extension to be typed that is composed of special characters, the user employs the number sign followed by the octal representation of the SIXBIT filename or extension. For example, #120000000000 represents the file named \*. If letters or digits are part of the filename or the extension containing the special characters, the octal representation of the letters or digits must also appear following the number sign. In other words, the number sign must be placed at the beginning of the filename and all characters following must be represented in octal.

The programs that recognize the number sign as described above are DUMP, DIRECT, and QUEUE. Refer to the PIP Programmer's Reference Manual for PIP's interpretation of the number sign.

**1.4.2.5 Date-Time Arguments** – Certain switches (e.g.,/BEFORE,/SINCE) require an argument with a date-time format(1). This argument can be either relative or absolute. A relative argument specifies a certain length of time from the current date and time, and an absolute argument specifies a particular date and time.

---

1

At the present time, DIRECT Version 3 is the only program that accepts the date-time format described in this section.

A relative argument is in the form

number of daysD:hours:minutes:seconds

where

number of days is optional.

D is required if number of days is given.

hours are optional if number of days (and the letter D) is given. Otherwise, hours are required.

minutes are optional.

seconds are optional. However, if seconds are used, minutes must be given or else seconds will be taken as minutes.

The colon is required if more than one field is given. A relative argument may be preceded by an optional sign (+ or -). If a sign is specified, it implies either past (-) or future (+). When the number of days is not specified, the time must be preceded by a + or -.

As an example,

-3D:4:27:21

means three days, four hours, 27 minutes, and 21 seconds ago. Similarly,

+4

means four hours from now.

An absolute argument is in the form

weekday or date:hours:minutes:seconds

where

weekday is the day of the week (or a unique truncated abbreviation) and is optional.

date is optional and has one of the following forms:

day of month-month-year (21-OCT-72)  
month-day of month-year (OCT-21-72)  
numeric month-day of month-year (10-21-72)

The month can be truncated to a unique abbreviation. The year (and its preceding hyphen) is optional and, if given, can be one, two, or four digits.

hours are based on 24-hour time and are required.

minutes are optional.

seconds are optional.

As an example,

WED:09:15:6

means before or since last Wednesday at 9:15:6 a.m.

Since the date is known to be past or future from either the switch used (/BEFORE and /SINCE imply past; /AFTER implies future) or by the plus or minus sign, an unspecified field is filled in so that the result is the next or last occurrence of the specified date. Thus, if the time is omitted, it is taken as 00:00:00 (midnight) if past, and 23:59:59 (11 o'clock, 59 minutes, and 59 seconds p.m.) if future. Thus,

/AFTER:SAT is after 23:59:59 next Saturday

/BEFORE:25-FEB is before last February 25th.

/SINCE:JUL-3-70:12 is since July 3, 1970 at noon.

### 1.4.3 Option Files

The user can create a file on his disk area into which he can place switches for the DIRECT and DUMP programs. The purpose of this file is to allow the user to automatically override the system default switches for individual programs by allowing him to specify his own defaults.

This file which must be called DSK:SWITCH.INI and must reside in the disk area under the user's UFD, can contain two types of lines. The first type is in the form

prog name/switch/switch...

where prog name is either DIRECT or DUMP and/switch is valid switch to the named program (e.g., DIRECT/DETAIL/NOSUMMARY). Only one line per program can exist in the file; however, this line can be continued on the next line by placing a hyphen as the last non-blank non-comment character before the carriage return-line feed sequence. When the user runs the named program, the switches will be used as the defaults for the program instead of the system-defined defaults. When the program is called, it searches for a file called SWITCH.INI. If such a file is not found, the system defaults for that program are used. If the file is found but a line for the program is not found, the system defaults are again used. When a line exists for the program, the switches in that line are used. The user can override a particular switch in the file by issuing a command string to the program containing a complement of the switch in the file. In other words, the switches in SWITCH.INI supply defaults for switches not specified by the user but are ignored if the user explicitly specified a switch.

The second type of line is in the form

prog name:option name/switch/switch...

where option name is the same as used on the /OPTION switch (refer to DIRECT and DUMP program descriptions). This type of line is used to override both the system defaults and the user's specified defaults. The user references a line of this type by including the /OPTION:option name switch in the command string to the program. If the option name



§

used on the switch does not appear in the file, the program outputs a warning message and uses the system defaults for that program.

Assume that the user creates a file called SWITCH.INI that contains the following lines:

```
DIRECT/FAST/UNITS/SUMMARY  
DIRECT:THISRUN/WORDS/ACCESS:25
```

If the user types DIR ,he receives a fast listing showing both the actual unit names instead of the structure names and the summary line. When he desires the normal directing listing instead of the fast listing, he must type a command string to DIRECT which includes the /NORMAL switch. Note that the disk unit names instead of the structure names and the summary line will still be output. To automatically list the length of the files in words instead of blocks and to update the access date of files 25 words or less, the user employs the /OPTION switch in the DIRECT command string as follows:

```
DIRECT/OPTION:THIS RUN
```

## 1.5 COMPIL-CLASS COMMANDS

Certain monitor commands simplify communication between the user and the system programs of the DECsystem-10 by allowing the user to type a short, concise monitor command string that causes a series of operations to be performed. These commands are known as COMPIL-class commands and are described in detail in Chapter 2. These commands cause the monitor to run the COMPIL program, which deciphers the command and constructs new command strings for the system program (e.g., TECO, PIP, LINED, FORTRAN) that actually processes the command. Each time CREATE, MAKE, EDIT, or TECO is executed, the command with its arguments is written as a temporary file in core or on the disk. Therefore, the file specification last edited may be recalled for the next edit without specifying the arguments again. (This is an exception to the requirement that the filename must always be specified.) For example, if the command

```
.CREATE PROGX.MAC
```

is executed, then the user may later type the command

```
.EDIT
```

instead of

```
.EDIT PROGX.MAC
```

assuming no other EDIT-class command that changed the filename was used in the interim.

The COMPILE, LOAD, EXECUTE, and DEBUG commands with their arguments are also written in a temporary file so that the file specification given last may be recalled without specifying the arguments again.

The temporary files containing these file specifications have filenames of the following form:

```
nnnxxx.TMP
```

where nnn is the user's job number in decimal, with leading zeros to make three digits, and xxx specifies the use of the file. Refer to Appendix C for a list of the temporary files.

### 1.5.1 Indirect Commands (@ Construction)

When there are many program names and switches, they can be put into a file and do not have to be typed in for each compilation. This is accomplished by the use of the @ file construction, which may be combined with any COMPIL-class command.

The @ file may appear at any point after the first word in the command. In this construction, the word file must be a filename, which may have an extension and a project-programmer number. If the extension is omitted, a search is made for the command file with the extension .CMD and then for a command file with a null extension. The information in the specified command file is then put into the command string to replace the characters @ file.

For example, if the file FLIST contains the string

```
FILEB,FILEC/LIST,FILED
```

then the command

```
.COMPILE FILEA,FILEB,FILEC/LIST,FILED,FILEZ
```

could be replaced by

```
.COMPILE FILEA,@FLIST,FILEZ
```

Command files may contain the @ file construction to a depth of 17 levels. If this process of indirection results in files pointing in a loop, the maximum depth is rapidly exceeded and an error message is produced.

The following rules apply in handling format characters in a command file.

1. Spaces are used to delimit words but are otherwise ignored. Similarly, the characters TAB, VTAB, and FORM are treated like spaces.
2. To allow long command strings, command terminators (CARRIAGE RETURN, LINE FEED, ALTMODE) are ignored if the first nonblank character after a sequence of command terminators is a comma. Otherwise, they are treated either as commas by the COMPILE, LOAD, EXECUTE, and DEBUG commands or as command terminators by all other COMPIL-class commands.
3. Blank lines are completely ignored.
4. Comments may be included in command files by preceding the comment with a semicolon. All text from the semicolon to the end of the line is ignored.
5. If command files are sequenced, the sequence numbers are ignored.

### 1.5.2 The + Construction (1)

A single relocatable binary file may be produced from a collection of input source files by the '+' construction. For example: a user may wish to construct a single program from several input files. If one input file is named FIRST.MAC, another named MIDDLE.MAC, and a third named LAST.MAC, the user can specify the command string

```
.COMPILE FIRST + MIDDLE + LAST
```

to obtain a single binary file from the three source files. This construction allows one input file to be used as part of several different compilations. For example, the file FIRST.MAC can later be used with SECOND.MAC and THIRD.MAC to obtain a different binary. Thus, the '+' construction permits a user to maintain in a single file material that is common to more than one compilation.

The name of the last input file in the string is given to any output (.REL, .CRF, and/or .LST) files (e.g., LAST and THIRD in the preceding examples). The source files in the '+' construction may each contain device and extension information and project-programmer numbers.

### 1.5.3 The = Construction (1)

Usually the filename of the relocatable binary file is the same as that of the source file, with the extension specifying the difference. This can be changed by the '=' construction, which allows a filename other than the source filename to be given to the associated output files. For example: if a binary file named BINARY.REL is desired from a source program named SOURCE.MAC, the following command is used.

```
.COMPILE BINARY=SOURCE
```

This technique may be used to specify an output name to a file produced by the use of '+' construction. To give the name WHOLE.REL to the binary file produced by PART1.MAC and PART2.MAC, the following is typed.

```
.COMPILE WHOLE=PART1+PART2
```

Although the most common use of the '=' construction is to change the filename of the output files, this technique may be used to change any of the other default conditions. The default condition for processor output is DSK:source.REL[self]. For example: if the output is desired on DTA3 with the filename FILEX, the following command may be used:

```
EXECUTE DTA3:FILEX=FILE1.F4
```

---

<sup>1</sup>

Used in COMPILE, LOAD, EXECUTE, and DEBUG commands only.

### 1.5.4 The < > Construction (1)

The <> construction causes the programs within the angle brackets to be assembled with the same parameter file. If + is used, it must appear before the <> construction. For example to assemble the files LPTSER.MAC, PTPSER.MAC, and PTRSER.MAC, each with parameter file PAR.MAC, the user could type

```
.COMPILE PAR+LPTSER, PAR+PTPSER, PAR+PTRSER
```

With the angle brackets, however, the command becomes

```
.COMPILE PAR+<LPTSER,PTPSER,PTRSER>
```

However, the following command is invalid:

```
.COMPILE <LPTSER,PTPSER,PTRSER>+PAR
```

### 1.5.5 Compile Switches

The COMPILE, LOAD, EXECUTE, and DEBUG commands can be modified by including switches in the command string. These switches can be used to indicate the processor to be used, to force a compilation, to generate listings, to create libraries, to search user libraries, and to obtain loader maps. Each switch is preceded by a slash and terminated with a non-alphanumeric character, usually a space or a comma. The switch used can be abbreviated if the abbreviation uniquely identifies the switch.

The switches used with these four commands are either temporary or permanent. A temporary switch applies only to the file immediately preceding it. An intervening space or comma cannot separate the filename and the switch. For example,

```
.COMPILE PROG,TEST/MACRO,SUBLET
```

The /MACRO switch applies only to the file named TEST.

A permanent switch applies to all files following it until modified by a subsequent switch. It is separated from the file by spaces, commas, or a combination of both. For example,

```
.COMPILE PROG/MACRO TEST,SUBLET  
.COMPILE PROG,/MACRO,TEST,SUBLET  
.COMPILE PROG,/MACRO TEST,SUBLET  
.COMPILE PROG/MACRO,TEST,SUBLET
```

In all four examples, the /MACRO switch applies to the files named TEST and SUBLET.

The switches that can be used with the COMPILE, LOAD, EXECUTE, and DEBUG commands are described in the individual command explanations in Chapter 2.

---

1

Used in COMPILE, LOAD, EXECUTE, and DEBUG commands only.

### 1.5.6 Standard Processor

Files with recognizable processor extensions (e.g., .MAC, .CBL, .F4, .ALG) are always translated by the processor implied by the extension. (1) For example, a file named DATPRO.CBL will be processed by the COBOL compiler. Files without a recognizable processor extension are compiled or assembled according to the standard processor, which is normally FORTRAN at the beginning of the command string. The user can control the setting of the standard processor by including switches in the COMPILE, LOAD, EXECUTE, or DEBUG command string. Refer to the appropriate command descriptions in Chapter 2 for the switches used to change the standard processor.

In the following examples, the installation has chosen FORTRAN as the standard processor. The command

```
.COMPILE NOEXT
```

causes the file named NOEXT (with a null extension) to be compiled by FORTRAN. The command

```
.COMPILE FILEZ.MIN
```

also compiles the file with FORTRAN since .MIN is not recognized as a processor extension. The command

```
.COMPILE APART,DATA/COBOL, TEST
```

causes the files APART and TEST to be compiled by FORTRAN and the file DATA by COBOL.

The switches used to change the standard processor can be temporary or permanent switches (refer to Paragraph 1.5.5). For example,

```
.COMPILE APART,/COBOL DATA,TEST
```

causes APART to be compiled by FORTRAN, and DATA and TEST to be compiled by COBOL.

Note that if source files are specified with the appropriate extensions, the subject of the standard processor can be disregarded, since files with processor extensions are always translated by the processor implied.

### 1.5.7 Processor Switches

Occasionally it is necessary to pass switches to the assembler or compiler being used in a COMPILE, LOAD, EXECUTE, or DEBUG command. For each translation (assembly or compilation), the COMPIL program sends a command string to the translator containing three parts: a binary output file, a listing file and the source files. To include switches with these files, the user must:

---

1

By setting the appropriate assembly switches, SNOBOL, BLISS, and MACX11 (the PDP-11 assembler for the PDP-10) will be recognized as processors. However, these assembly switch settings are not supported.

1. If the + construction is used, group the switches according to each related source file.
2. Group the switches according to the three types of files for each source file. The order of the groups of switches is binary, listing and source.
3. For each source file, separate the groups of switches by commas.
4. Enclose all the switches for each source file within one set of parentheses.
5. Place each parenthesized string immediately after the source file to which it refers.

The COMPIL interprets the groups of switches according to the following rules.

1. Switches before closing right parenthesis are source. (SSSS)
2. Switches before first comma are binary. (BBBB)
3. Switches before second comma are listing. (LLLL)
4. The order of the switches is: BBBB, LLLL, SSSS

For example:

(SSSS)	source switch
(BBBB)	binary switch
(BBBB,.)	binary switch
(BBBB, LLLL, SSSS)	binary, list, source switches
(,SSSS)	source switch(es)
(,,SSSS)	source switch(es)
(,LLLL,.)	listing switch(es)
(BBBB,SSSS)	binary, source switches
(BBBB,LLLL,.)	binary, list switches
(,LLLL,SSSS)	list, source switches

The processor switches are listed in Table 1-2, along with their meanings and the types of files to which they apply.

Examples:

- | .DEBUG TEST(N,) Suppress typeout of errors during assembly.
- .COMPILE OUTPUT=MTA0:(S,M,W)/L Rewind the magtape (W), compile the first file, produce binary output for the K110(S), and include the MACRO coding in the output listing (M). Output files are given the names OUTPUT.REL and OUTPUT.LST.
- .COMPILE/MACRO A=MTA0:(,Q,W)/L Rewind the magtape (W), compile the first file, and suppress Q (questionable) error indications on the listing. Note that when a binary switch is not present, the delimiting comma must appear.
- .COMPILE/MACRO A=MTA0:(,Q,)/L Compile file at current position of the tape and suppress Q error indications on the listing. Note that when the source and binary switches are not present, the delimiting commas must appear.

Table 1-2  
Processor Switches

Processor	Binary	Listing	Source	Meaning	
ALGOL	D	N		Set dynamic storage region for own arrays (known as the heap).	
			E	The source file has line numbers in columns 73-80.	
			L	List the source program. Suppress the error print out on the terminal.	
			Q	Delimit the words in quotes.	
			S	Suppress the listing of the source program.	
COBOL	A	A	A	Allow the listing of code generated.	
		C		Produce a cross-referenced listing of all user-defined items in the source program.	
	E	E		E	Check the program for errors but do not generate code.
				I	Suppress generation of the starting address of a main program.
	J				Force a starting address to be generated for a subprogram.
				L	Use the preceding file descriptor as a library file whenever the COPY verb is encountered.
	M	M	N	M	Print a map showing the parameters of the user-defined item.
					Suppress output of source errors on the terminal.
	P	R			Do not generate trace calls and symbols.
S	S		S	The source file has sequence numbers in columns 1-6 and comments starting in column 73.	
					Rewind the magnetic tape.
W	Z	W	W	Zero the DECTape directory.	
				Z	



Table 1-2 (Cont.)

## Processor Switches

Processor	Binary	Listing	Source	Meaning	
FORTRAN	A B	A	A	Advance magnetic tape reel by one file. Backspace magnetic tape reel by one file.	
		B	B		
	S  T  W Z	C		Generate a CREF-type cross-reference listing.	
		D		List error message codes only.	
		E		Print an octal listing of the binary program in addition to the symbolic listing. Must be accompanied by /M.	
		I		Translate the letter D in column 1 as a space and treat the line as a normal FORTRAN statement.	
		M		Include MACRO coding in output listing.	
		N		Suppress output of error messages on the terminal.	
		T	T	T	Produce code for execution on the KA10 if running on the KI10, and vice-versa.
		W	W	W	Skip to the logical end of magnetic tape.
Z	Z		Rewind the magnetic tape. Zero the DECtape directory.		
FORTRAN-10		C		Generate a file that can be input to the CREF program.	
		E		Include the octal formatted version of the object file in the listing.	
		I		Translate the letter D in column 1 as a space and treat the line as a normal FORTRAN statement.	
		KA		Compile the program, producing code to execute on the KA10 processor.	
		KI		Compile the program, producing code to execute on the KI10 processor.	
		M		Add the mnemonic translation of the object code to the listing file.	
		NOE	NOE		Suppress output of error messages to the terminal.
		NOW	NOW		Suppress output of warning messages to the terminal.
		O			Perform optimization of global symbols when producing processor code.
		S		S	Perform compilation checking for syntax errors only.

Table 1-2 (Cont.)

## Processor Switches

Processor	Binary	Listing	Source	Meaning
MACRO	A	A	A	Advance magnetic tape reel by one file.
	B	B	B	Backspace magnetic tape reel by one file.
		C		Produce listing file in a format acceptable as input to CREF.
		E		List macro expansions.
		F		Byte sizes match the format of the instruction.
		G		Byte sizes are two 18-bit fields.
		L		Reinstate listing (used after list suppression by S switch).
		M		Suppress ASCII text in macro and repeat expansion (SALL).
		N		Suppress error printouts on the terminal.
	O	O	O	Allow literals to occupy only one line.
	P			Increase the size of the pushdown list.
	Q	Q	Q	Suppress questionable (Q) error indications on the listing.
		S		Suppress listing.
	T	T	T	Skip to the logical end of magnetic tape.
	W	W	W	Rewind the magnetic tape.
		X		Suppress all macro expansions.
	Z	Z	Zero the DECTape directory.	

**1.5.8 Switches to the Linking-Loaders**

Two linking-loaders are available on the DECsystem-10: the LOADER program and the LINK-10 program. (Refer to the LOADER Programmer's Reference Manual, DEC-10-LLZB-D, and the LINK-10 Programmer's Reference Manual, DEC-10-ULKMA-A-D, for complete information on these programs.) In complex loading procedures, it may be necessary to pass switches to one of the linking-loaders in order to direct its operation. The passing of switches is accomplished by the percent (%) character.

**1.5.8.1 Passing LOADER Switches**

The % character takes an optional leading sign (+ or -) and one letter (or a sequence of digits and one letter) following it. Therefore, to set a program origin of 6000 for file PROG3, the user types

```
.LOAD PROG1,PROG2,%6000 OPROG3.PROG4
```

The %60000 switch, with the % character replaced with the /character, is passed to the LOADER. The COMPIL program allows more than one LOADER switch to be specified. For example,

```
.LOAD PROG %F/MAP
```

Refer to the LOAD command in Chapter 2 for a description of the /MAP switch.

### 1.5.8.2 Passing LINK-10 Switches

Following the % character is the LINK-10 switch specification preceded and followed by a delimiter. The delimiter can be any character; however, the user must be careful that the character he uses does not have a specific meaning to COMPIL. For example, the @ character indicates an indirect command file, and the semicolon causes the remainder of the line to be treated as a comment and thus ignored. The recommended delimiter is a single or double quote character. The beginning and ending delimiter must be the same character.

A LINK-10 switch specification consists of the switch name and optionally a keyword and a value. The items in the specification are separated by colons. Thus, to set a program origin of 6000 for file FILEB, the user types

```
.LOAD/LINK FILEA, % 'SET:;LOW.;6000'FILEB,FILEC
```

Refer to the LOAD command in Chapter 2 for a description of the /LINK switch.



## CHAPTER 2

# SYSTEMS COMMANDS AND PROGRAMS

Although there is one operating system for all configurations of the DECsystem-10, some commands may not be included in each DECsystem-10. This is especially true of the DECsystem-1040, the basic system intended for small installations that do not want all of the system's features because of a constraint on core. Commands are deleted from the DECsystem-1040 by feature test switches (recognized by the beginning characters FT) defined at MONGEN time. In the standard DECsystem-1040, many of these switches are not set and, therefore, the corresponding commands are not available. This saves core but limits various features of the operating system. In the command descriptions that follow, the Characteristics section indicates if the switch is normally off in the DECsystem-1040. If not stated, the command is available on all DECsystem-10s.

In many cases, there are two commands to run a program. For example, the indirect command MAKE and the direct command R TECO both run the TECO program. In the DECsystem-1040, the switch implementing the indirect command may not be set but the switch implementing the direct command is always set. Therefore, it is always possible to run a program with the .R or .RUN command, even if the switch implementing the corresponding indirect command is off.

### 2.1 COMMANDS BY FUNCTIONAL GROUPS

Although the commands are arranged in alphabetical order for ease of reference, they can be divided into functional groups for ease of learning. These groups with their associated commands are as follows.

#### 2.1.1 Job Initialization Commands

Since the system is limited to authorized persons, these commands protect the system from unauthorized use.

INITIA	Performs standard system initialization for the terminal issuing the command.
LOGIN	Gains access to the system.

#### 2.1.2 Facility Allocation Commands

The monitor allocates peripheral devices, file structure storage, and core memory to users on request and protects these allocated facilities from interference by other users. Software provisions are incorporated in the monitor to differentiate the central stations from the remote stations. Certain monitor commands, for example, ASSIGN and PLEASE, include station

identification arguments to allow both user-access and allocation of system resources at any station. This feature gives the user considerable flexibility in allocating system facilities and directing input and output to the station of his choice. For example, by specifying a station number, the user can assign devices and input data from a peripheral device at a station other than his own. In addition, by using the LOCATE command, he can logically establish his job at a station other than his physical station. If the station identification argument is not included in a command, the system automatically directs input and output to the user's logical station. The user's logical station is the same as his physical station if he has not issued the LOCATE command.

When a nonsharable device is assigned to a job, it is removed from the monitor's pool of available resources. Any attempt by another user to reference or assign the device fails. Thus, a user should never leave the system without first returning his allocated facilities to the monitor pool. Allocated facilities are automatically returned to the monitor pool when the user deassigns them or kills his job. Until a user returns these facilities, no other users may utilize them except through operator intervention.

Assignable devices (i.e., nondisk and nonspooled devices) in the monitor's pool of available resources are designated as being either unrestricted or restricted devices. An unrestricted device can be assigned (ASSIGN command or INIT UUC) by any user. A restricted device can be assigned only by a privileged job (i.e., a job logged in under [1,2] or running with the JACCT bit set). However, a nonprivileged user can have a restricted device assigned to him via the MOUNT command. This command allows operator intervention for the selection or denial of a particular device; thus the operator can control the use of the assignable devices. This is particularly useful when there are multiprogramming batch and interactive jobs competing for the same devices. The restricted status of a device is set or removed by the OPSER commands :RESTRICT and :UNRESTRICT.

The facility allocation commands are as follows:

ASSIGN	Allocates an I/O device to a user's job without operator intervention.
CLOSE	Terminates I/O in progress on specified device but does not release the device.
CORE	Types or modifies the amount of core assigned to the user's job.
DEASSIGN	Returns devices assigned to the user's job to the monitor's pool of available devices and clears logical names.
DISMOUNT	Returns, via the operator, devices assigned to the user's job to the monitor's pool of available devices.
FINISH	Terminates I/O in progress on the specified device, performs the RELEASE UUC and the DEASSIGN command.
LOCATE	Establishes the user's job at the specified station.
MOUNT	Allocates an I/O device to the user's job via the operator.
REASSIGN	Gives the specified device to the designated job.

SET BLOCKSIZE	Sets the default blocksize for the specified magnetic tape unit.
SET CDR	Sets the filename for the next card reader spooling interrupt.
SET CPU	Sets the CPU specification (i.e., CPxn, NO CPxn, ALL, ONLY CPxn) for the job.
SET DENSITY	Sets the default density for the specified magnetic tape unit.
SET DSKPRI	Sets the priority for the job's disk operations (data transfers and head positioning).
SET HPQ	Sets the high priority scheduler run queue for the job.
SET SPOOL	Adds devices to or deletes devices from the list of spooled devices for the job.
SET TTY (or TTY)	Sets properties to be associated with the terminal.

### 2.1.3 Source File Preparation Commands

These commands call the system editing programs in order to create or edit a specified text file. The system editing programs available are LINED (a line-oriented editor) and TECO (a character-oriented editor). In general, the editor used to create the file should be used for editing, since LINED requires line-blocked files and TECO does not.

CREATE	Makes new file on disk for creation with LINED.
EDIT	Opens the specified file, already existing on disk, for editing with LINED.
MAKE	Opens a new file on disk for creation with TECO.
TECO	Opens a specified file that already exists for editing with TECO.

### 2.1.4 File Manipulation Commands and Programs

The commands and programs in this group allow the user to manipulate his files to any desired extent. He can list source files, and DEctape and disk directories on the terminal or the line printer, possibly via the spooling mechanism. He can delete or rename files from disk and DEctape. In addition, the user can transfer files between standard I/O devices, perform conversion between various core image formats, and read and write various directory formats. Disk space can be either allocated for a new file or reallocated for an existing file. Finally, the user can place files in the system queues and obtain listings of entries in those queues.

ALCFIL	Allocates space for a new file or reallocates space for an existing file.
BACKSPACE	Spaces a magnetic tape backward the specified number of files or records.

CPUNCH	Places entries into the card punch output spooling queue.
DELETE	Deletes files from DECTape or disk.
DIRECT	Lists the directory entries for the specified arguments.
DTCOPY	Transfers files from one device to another; copies contents of one DECTape to another.
EOF	Writes an end-of-file mark on the specified magnetic tape.
FAILSAFE	Saves and restores disk files.
FILCOM	Compares two versions of a file and outputs any differences.
FILE	Provides remote control via the operator for DECTape-to-disk and disk-to-DECTape transfers.
FILEX	Converts between various core image formats, and reads and writes various directory formats on DECTape.
GLOB	Reads multiple binary files to produce an alphabetical cross-referenced listing of all global symbols encountered.
LABEL	Writes an identifier onto a DECTape.
LIST	Writes the specified files on the line printer.
PLOT	Places entries into the plotter output spooling queue.
PRESERVE	Renames the specified files with the standard protection inclusively ORed with 100.
PRINT	Places entries into the line printer output spooling queue.
PROTECT	Renames the specified files with the requested protections.
PUNCH	Places entries into the paper tape punch output spooling queue.
QUEUE	Enters items into the specified system queue.
RENAME	Changes one or more items of the file specification of files on DECTape or disk.
REWIND	Rewinds a magnetic tape or DECTape.
SKIP	Moves the specified magnetic tape forward the designated number of files or records or to the logical end of the tape.
SUBMIT	Places entries into the Batch input queues.
TPUNCH	Places entries into the paper tape punch output spooling queue.



TYPE	Types the specified files on the user's terminal.
UNLOAD	Rewinds and unloads the specified magnetic tape or DECTape.
ZERO	Clears the directory of the specified device.

### 2.1.5 Object Program Preparation Commands and Programs

The commands and programs in this group are used to prepare object programs and save the user's core area as one or two files.

COMPILE	Produces relocatable binary files (.REL extensions) for the specified source files.
CREF	Lists on LPT: any cross-referenced listing files generated by a previous COMPILE, LOAD EXECUTE, or DEBUG command.
DEBUG	Produces relocatable binary files for the specified source files, loads the .REL files, and prepares for debugging with DDT.
EXECUTE	Produces relocatable binary files for the specified source files, loads the .REL files, and begins execution.
FUDGE	Creates a library .REL file by reading a temporary file generated by a previous COMPILE, LOAD, EXECUTE, or DEBUG command.
FUDGE2	Updates files containing relocatable binary programs, and manipulates programs within program files.
LOAD	Produces relocatable binary files for the specified files and loads the .REL files generated.
SAVE	Writes a core image of the user's core area on the specified device.
SSAVE	Writes a core image of the user's core area on the specified device with the high segment.

### 2.1.6 Object Program Control Commands

By using the commands in this group, the user can load core image files from retrievable storage devices (i.e., disk, DECTape, magnetic tape). These files can be retrieved and controlled from the user's terminal. Files stored on disk and DECTape are addressable by name. Files on magnetic tape require the user to pre-position the tape to the beginning of the

file. Refer to DECSYSTEM-10 Monitor Calls, Chapter 1, for a description of the job data area locations referenced by the command descriptions in this group.

CCONT	Continues the program from the point at which it was interrupted, but leaves the terminal in monitor mode.
CONT	Continues the program from the point at which it was interrupted.
CSTART	Begins execution of a program that was either loaded with a GET command or interrupted, but leaves the terminal in monitor mode.
DDT	Copies the saved program counter and then starts.
GET	Loads a core image from the specified device, but does not begin execution.
HALT	Stops the job and stores the program counter in the job data area.
JCONT	Continues the specified job if it was waiting for operator intervention.
R	Loads a core image from the system device (SYS:) and starts it at the location specified within the file.
REENTER	Starts the program at an alternate entry point specified by the program.
RUN	Loads a core image from the specified device and starts it at the location specified within the file.
SET DSKFUL	Controls the job when the user has exhausted his disk space.
START	Begins execution of a program either previously loaded with the GET command or interrupted.

### 2.1.7 Object Program Examination Commands

The commands in this group aid the user in examining and analyzing his object program. Dumps of the user's core area can be taken and later processed by the system program DUMP according to the arguments specified by the user.

D(deposit)	Deposits information in the user's core area.
DCORE	Writes a core image file of the user's core area.
DUMP	Writes a core image file and analyzes the file written in order to provide printable output.
E(examine)	Examines the specified core location in the user's area.

### 2.1.8 Multiple Job Control Commands and Programs

There is not necessarily a one-to-one relationship between jobs and terminals. A terminal must initiate a job, but the user or operator may issue commands to permit a job to float in a detached state where it is not associated with a particular terminal. Thus, more than one job may be controlled from the same terminal.

ATTACH	Detaches the current job and connects the terminal to the specified detached job.
CCONT	Continues the program from the point at which it was interrupted, leaving the terminal in monitor mode.
CSTART	Begins execution of a program that was either loaded with a GET command or interrupted, leaving the terminal in monitor mode.
DETACH	Disconnects the terminal from the current job without affecting the status of the job.
OPSER	Provides multiple job control from a single terminal.
REATA	Transfers the job from the current terminal to the specified terminal.

### 2.1.9 Job Termination Command

When the user leaves the system, all facilities allocated to his job must be returned to the monitor facility pool so that they are available to other users.

KJOB	Gives up access to the system.
------	--------------------------------

### 2.1.10 Sending Messages

The commands in this group allow the user interconsole communications with other users of the system or with operators at any station. In addition, the user may record information in a disk file to be read by the operations staff at a later time.

GRIPE	Accepts text from a user and records it in a disk file for the operations staff.
PLEASE	Provides two-way communication between the user and the operator.
SEND	Provides a one-way line of communication.

### 2.1.11 Job Information Commands and Programs

The user can obtain various job-related information with this group of commands. This information includes the number of his job, the quotas for each file structure associated with his job, and the running time and disk space that his job has used. In addition, the user may type or modify his file structure search list.

DSK	Types disk usage for the combined structures of the specified job.
PJOB	Outputs the job number to which the terminal is currently attached.
QUOLST	Types the used, logged-in quota, and logged-out quota for each file structure for which the user has access, followed by the number of free blocks left on that structure.
SETSRC	Manipulates the job's search list or system's search list.
SET TIME	Sets the central processor time limit for the job.
SET WATCH	Sets the output of incremental job statistics.
TIME	Outputs the running time for the specified job.
USESTAT	Outputs status information pertaining to the current job.

### 2.1.12 System Information Commands and Programs

With the commands in this group, the user is able to obtain system status information, including the time of day, the list of available devices, file structures, and physical units not in file structures, the scheduled use of the system, and the location of a specific peripheral device.

DAYTIME	Types the current date followed by the time of day.
RESOURCES	Outputs the names of all available devices (except for TTYs and PTYs), all file structures, and all physical units not in file structures.
SCHED	Outputs the schedule bits set by the operator.
SYSTAT	Prints system information about the current system.
VERSION	Outputs the version number of a program.
WHERE	Outputs the station number at which the specified device is located.

## Function

The ALCFIL program enables the user to allocate space for a new file or reallocate space for an existing file in one contiguous region on the disk. The size of the region is restricted by the size of the cluster count field (usually 512) times the cluster size of the file structure times the number of pointers in a disk device data block (not less than 10).

## Command Format

### R ALCFIL

The ALCFIL program responds with

```
/H FOR HELP  
FILE?
```

The user may respond with

```
dev:file.ext [proj,prog]  
or /H (for help)  
or /X (to exit)
```

where dev: is a file structure or physical unit name. If dev: is omitted, DSK is assumed. If one of the other arguments is omitted, 0 is assumed. If a filename is specified, the number of blocks presently allocated, if nonzero, is typed. ALCFIL responds with

```
ALLOCATE?
```

User may type N or N,M (decimal numbers)

```
N = total number of blocks to be allocated for the file.  
M = logical block within the file structure or unit (depending on dev:)  
where the allocation is to begin.
```

If the total number of blocks requested cannot be allocated (because of disk quotas), a partial allocation is given and the message

```
PARTIAL ALLOCATION ONLY
```

is typed. The user can issue the DIRECT command with the ALLOCATE switch to determine the number of blocks allocated. If the new blocks can be allocated, the message

```
ALLOCATED
```

is typed.

Since an extended ENTER (refer to DECSYSTEM-10, Monitor Calls) is executed to allocate the new blocks, the file need not exist before the blocks are allocated.

### Characteristics

The R ALCFIL command:

Places the terminal in user mode.

Runs the ALCFIL program, thereby destroying the user's core image.

### Associated Messages

Refer to Chapter 4.

### Example

```
._R ALCFIL )  
./H FOR HELP  
FILE? TEST4.TST )  
ALLOCATE? 2000 )  
  
ALLOCATED  
FILE? TEST5.TST )  
ALLOCATE? 1000 )  
  
ALLOCATED  
FILE? TEST5.TST )  
1000 BLOCKS ALREADY ALLOCATED  
ALLOCATE? 500 )  
  
ALLOCATED  
FILE? DSKB:FILEA  
ALLOCATE? 3000 )  
  
PARTIAL ALLOCATION ONLY  
FILE?  
  
EXIT  
  
._DIR/ALLOC )  
FILEA 175 <057> 12-APR-74 DSKB  
  
.
```

## Function

The ASSIGN command allocates an I/O device to the user's job for the duration of the job or until a DEASSIGN command is given. This command, applied to DECtapes, clears the copy of the directory currently in core, forcing any directory references to read a new copy from the tape. (Refer to DECsystem-10 Monitor Calls for further details.)

Although DECtape is the only device that should be ASSIGNED before use, to ensure that the monitor has a copy of the proper DECtape directory in its core area, it is wise to ASSIGN all devices, such as magnetic tape, before use.

## Command Formats

### 1. ASSIGN phys-devn log-dev

phys-devn = any physical device listed in Table 1-1 in Paragraph 1.4.2.2, followed by a 1-to-3 digit number representing a specific unit, or any file structure name. This argument is required. With this command format, the monitor attempts to assign the device specifically requested. If unable to assign the device, the monitor types an appropriate message (refer to Chapter 4).

log-dev = a logical name assigned by the user. This argument is optional. Except for disk devices, only one logical name can be assigned to a physical device. Subsequent ASSIGN commands to all devices except disk devices replace the old logical name with the new one. Logical names are disassociated from all devices by the DEASSIGN command.

### 2. ASSIGN phys-devSnn log-dev

phys-devSnn = any physical device followed by the letter S and a 1 or 2 digit number representing a specific station, or any file structure name. This argument is required. With this command format, the monitor attempts to assign a device at the requested station. An appropriate message is typed if the device cannot be assigned (refer to Chapter 4).

log-dev = same as above.

### 3. ASSIGN phys-dev log-dev

phys-dev = any physical device followed by a null argument implying any device of the designated type, or any file structure name. This argument is required. With this command format, the monitor attempts to assign the requested device at the user's logical station. If this type of device does not exist at the user's logical station, the monitor attempts to assign the device at

the central station. If unable to assign the device, the monitor types an appropriate message (refer to Chapter 4).

log-dev = same as above.

## Characteristics

The ASSIGN command:

Leaves the terminal in monitor mode.

## Restrictions

A comma may not be used to separate the logical and physical device names. If a comma is used, the monitor terminates its scan at the comma; therefore, the logical name is not assigned.

Non-privileged jobs (i.e., jobs not logged in as [1,2] or running with JACCT set) can only use this command to allocate unrestricted I/O devices. Restricted devices can be obtained by non-privileged jobs via the MOUNT command. The ASSIGN when issued by a privileged job allocates both restricted and unrestricted devices.

## Associated Messages

Refer to Chapter 4.

## Examples

```
.ASSIGN LPT2: )  
LPT2 ASSIGNED
```

The user assigns a specific line printer (LPT2).

```
.AS CDRS2: )  
CDR4 ASSIGNED
```

The user assigns any available card reader at station 2.

```
.ASSIGN TTY1:LPT: )  
TTY1 ASSIGNED
```

The user assigns TTY 1 and gives it logical name LPT.

```
.AS LPT: )  
LPT4 ASSIGNED
```

The user assigns any available line printer. The LPT chosen is at either the user's station or the central station.

```
.ASSIGN DTA2: )  
?DEVICE NOT ASSIGNABLE
```

A non-privileged user attempted to allocate a restricted device (DTA2).

```
.ASSIGN DTA: )  
DTA4 ASSIGNED
```

The user then uses the generic device name (DTA) to obtain the device. He could have used the MOUNT command to assign the restricted device DTA2.



## ATTACH Command

### Function

The ATTACH command detaches the current job, if any, and connects the terminal to a detached job.

### Command Format

ATTACH job [proj,prog]

job = the job number of the job to which the terminal is to be attached. This argument is required.

[proj,prog] = the project-programmer number of the originator of the desired job. This argument may be omitted if it is the same as the job to which the terminal is currently attached. The operator (device OPR) or a user logged-in under [1,2] may always attach to a job although another terminal is attached, provided he specifies the proper [proj,prog].

To prevent users from attaching the jobs without knowing the PASSWORD associated with the job, a new job is temporarily created when the [proj,prog] argument is specified. This temporary job runs LOGIN to check the password. This can result in the current job not being able to attach to the specified job if the job capacity of the system would be exceeded with the creation of the temporary job. However, the current job is still detached even if there are no available jobs. The operator or any job logged-in as [1,2] can always attach to another job since they do not require the creation of a temporary job.

### Characteristics

The ATTACH command:

- Leaves the terminal in monitor mode.

- Does not require LOGIN.

- Depends on FTATTACH which is normally absent in the DECsystem-1040.

### Restrictions

Remote users cannot attach to jobs with a project number of 1. Batch users cannot issue this command.

## Associated Messages

Refer to Chapter 4.

### Examples

1. .ATT 1 )  
FROM JOB 5

The user attaches to job 1 from job 5. The two jobs have the same [proj,prog] and therefore, neither the [proj,prog] nor password argument is required.

.
2. .LOG 27,235 )  
JOB 7 5S04 TTY25  
PASSWORD:  
1634 23-FEB-72 WED

The user logs-in and is given job 7. TTY25 is now attached to job 7.

.ATTACH 36 [50,27] )  
FROM JOB 7  
PASSWORD:

The user attaches to an existing job (36) and thereby detaches his current job (7). Since the [proj,prog] associated with job 36 is different than the user's, he must specify the [proj,prog] of the desired job. The system then requests the PASSWORD. If the given PASSWORD is correct, the terminal is attached to job 36.

The terminal is attached to job 36.

.

.ATTACH 7 )  
?CAN'T ATT TO JOB

The user attempts to attach to job 7. The command fails because the [proj,prog] of job 7 is not the same as the [proj,prog] of job 36. The terminal is still attached to job 36.

.K/F )  
JOB 36,USER [50,27] LOGGED OFF TTY25 1635 23-FEB-74  
RUNTIME OMIN,00,34sec

The user killed job 36. The terminal is currently not attached to any job.

.ATTACH 7  
?CAN'T ATT TO JOB

Since the terminal is currently not attached to a job, the command fails because there is no [proj,prog] to compare with the [proj,prog] of job 7.

.ATTACH 7 [27,235] )  
PASSWORD:

The command is accepted and the PASSWORD is requested. The message FROM JOBn is not output since the terminal was not attached to a job.

.

The terminal is attached to job 7.

## BACKSPACE Command (1)

### Function

The BACKSPACE command spaces a magnetic tape backward a specified number of files or physical records. This command, depending on its arguments, is equivalent to the following PIP command strings:

MTAn: (M#nB) –  
MTAn: (M#nP) –

SAVED files on magnetic tape always contain two files, a high segment file and a low segment file. If one of the segments is missing, a null file containing one record is written before the EOF for missing segment. Thus in order to space backward over a SAVED file, the user must backspace two files. Refer to Appendix D for the format of a SAVED file.

### Command Formats

1. BACKSPACE MTAn: x FILES  
skips backward x files.
2. BACKSPACE MTAn: x RECORDS  
skips backward x records.

### Characteristics

The BACKSPACE command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to chapter 4.

---

1

This command runs the COMPIL program, which interprets the command before running the PIP program.

Version 21A COMPIL  
Version 33 PIP

Examples

\_BAC MTA2: 7 RECORDS )

\_BACKSP MTA3: 11 FILES )

## CLOSE Command

### Function

The CLOSE command terminates any input or output currently in progress on the specified device, and automatically performs the CLOSE UUC (refer to DECsystem-10 Monitor Calls). Files are CLOSEd, but not RELEASEd, and logical names and device assignments are preserved. Since most programs CLOSE files when they finish performing a command string, the CLOSE command is provided for the occurrence of a program not terminating or a program being debugged. This command causes any disk files being written to be entered into the user's UFD. If a CLOSE is not done, the next RESET by a command (R, RUN, GET) or program will delete the partially written file.

### Command Format

CLOSE dev

dev = the logical or physical name of the device on which I/O is to be terminated. This argument is optional.

If dev is omitted, I/O is terminated on all devices, except for the job's controlling terminal, and all files are CLOSEd.

### Characteristics

The CLOSE command:

Leaves the terminal in monitor mode.

Requires core.

Depends on FTFINISH which is normally absent in the DECsystem-1040.

### Restrictions

The user cannot continue, but can start at the beginning of his program or enter DDT.

### Associated Messages

Refer to Chapter 4.

core if the monitor has the TMPCOR feature. Therefore, issuing one of these commands without any arguments causes the arguments saved in the temporary file to be reused (refer to Paragraph 1.5).

The COMPILE command accepts several command constructions: the @ construction (indirect commands), the + construction, the = construction, and the <> construction. Refer to Paragraph 1.5 for a complete description of each of these constructions.

## Command Format

### COMPILE list

list = a single file specification, or a string of file specifications separated by commas. A file specification consists of a device name, a filename with or without an extension, and a directory name (refer to Paragraph 1.4.2.4).

The following switches can be used to modify the command string. These switches can be temporary or permanent switches (refer to Paragraph 1.5.5). Note that all the switches allowed with the LOAD, EXECUTE, and DEBUG commands can be used with the COMPILE command. However, only the switches pertinent to COMPILE are listed below: the others are ignored.

/ALGOL	Compile the file with ALGOL. Assumed for files with the extension of .ALG.
/BIN	Generate a binary file, for each file compiled. The filename of the binary file follows the standard convention for determining the filename of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .REL. This is the default action.
/BLIS (1)	Compile the file with BLISS10. Assume for files with the extension of .B10 and .BLI.
/COBOL	Compile the file with COBOL. Assumed for files with the extension of .CBL.
/COMPILE	Force a compilation on this file even if a binary file exists with a newer date and time than the source file. This switch is used to obtain an extra compilation (e.g., in order to obtain a listing of the compilation) since normally compilation is not performed if the binary file is newer than the source file.

---

<sup>1</sup> BLIS10 will be recognized as a translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

**/CREF** Produce a cross-reference listing file on the disk for each file compiled for later processing by the CREF program. The filename of the listing file follows the standard convention for determining the filename of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .CRF. The file can then be listed with the CREF command. However, with COBOL files, the cross-referenced listing is always appended to the listing file. No additional command need be given to obtain the listing.

**/FORTRAN** Compile the file with a FORTRAN compiler. Assumed for files with the extension of .F4 and .FOR and all files with non-recognizable translator extensions if FORTRAN is the standard translator. This switch is needed if the file has a non-recognizable translator extension and FORTRAN is not the standard translator or is not the current default (e.g., COMPILE/ALGOL FILA, FILB, FILC/ FORTRAN).

**/FUDGE** Create a disk file containing the names of the .REL files produced by the command string. When the FUDGE command is given, PIP reads this file in order to generate a library REL file. (Refer to the FUDGE2 program description.) Argument to this switch are:

**/FUDGE:dev:file.ext [proj,prog]**

dev: - the device on which to write the file. If the device is omitted, DSK: is assumed.

file.ext - the name of the library file. The filename is required. If the extension is omitted, it is assumed to .REL.

[proj,prog] - the directory in which to place the file. The user's directory is assumed if none is given.

This switch is permanent in the sense that it pertains to all .REL files generated by the command string.

**/F10** Use the FORTRAN-10 translator when compiling the associated FORTRAN file. This should be used as a permanent switch (i.e., placed before any file specifications) if loading or execution of the command string will be requested because it is not possible to load F40 and FORTRAN-10 binary files together.

/F40	Use the F40 translator when compiling the associated FORTRAN file. This is the current default action. This should be used as a permanent switch (i.e., placed before any file specifications) if loading or execution of the command string will be requested because it is not possible to load F40 and FORTRAN-10 binary files together.
/KA10	Designate the machine on which the program will execute once it has been loaded. These switches are needed for FORTRAN-10 programs since the compiler generates different codes for the KA10 and KI10 processors. The /KA10 switch is the current default.
/KI10	
/LIST	Generate a disk listing file, for each file compiled. The filename of the listing file follows the standard conventions for determining the filename of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .LST. These files can be listed later with the LIST command. If the line printer is being spooled for this job, the listing files are written on device LPT and are automatically spooled at LOGOUT time. Unless this switch is specified, listing files are not generated.
/MACRO	Assemble the file with MACRO. Assumed for files with extension of .MAC.
/MACX11(1)	Assemble the file with MACX11. Assumed for files with extension of .P11.
/MANTIS(2)	Compile the program with the MANTIS debugging information. This switch affects FORTRAN files only.
/NEW	Run the appropriate language translator from the experimental system library (device NEW:) area [1,5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:. See the following NOTE.

---

<sup>1</sup> MACX11 (the PDP-11 assembler for the PDP-10) will be recognized as a translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

<sup>2</sup> MANTIS will be recognized as the debugging program only if the appropriate assembly switch is set. However, this switch setting is not supported.



/NOBIN	Do not generate binary files. Unless this switch is given, binary files are generated. This switch, when combined with the /CREF or /LIST switch, is useful when compiling programs solely for the purpose of generating listings.
/NOCOMPILE	Complement the /COMPILE switch by not forcing a compilation on a source file whose date is not as recent as the date on the binary file. /NOCOMPILE is the default action.
/NOLIST	Do not generate listing files. This is the default action.
/NOMANTIS(1)	Compile the program without the MANTIS debugging information. The switch affects FORTRAN files only.
/OLD	Run the appropriate language translator from the system library of old programs (device OLD:) which resides on the disk area [1,3]. If the translator does not exist on device OLD:, try to obtain it from device SYS:. See the following NOTE.
/SELF	Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for an individual who keeps a private copy of a translator in his own disk area. System programmers occasionally keep experimental versions of standard translators in their disk areas in order to test new features. See the following NOTE.
/SNOBOL(2)	Compile the file with SNOBOL. Assumed for files with an extension of .SNO.
/SYS	Run the appropriate language translator from the system library (device SYS:) area of [1,4]. This is the default action. See the following NOTE.

---

<sup>1</sup> MANTIS will be recognized as the debugging program only if the appropriate assembly switch is set. However, this switch setting is not supported.

<sup>2</sup> SNOBOL will be recognized as a translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

## NOTE

Once a language translator has been specified from a particular area (e.g., /NEW), it cannot be changed within the same command string; i.e., the following is illegal:

```
.COMPIL TESPGR.F4/NEW,SUBRTN.F4/SYS
```

However, the following is valid:

```
.COMPIL TESPGR.F4/NEW  
.COMPIL SUBRTN.F4/SYS
```

## Characteristics

The COMPILE command:

Leaves the terminal in monitor mode.

Runs the appropriate language translators, thereby destroying the user's original core image.

## Restrictions

The wildcard construction cannot be used.

## Associated Messages

Refer to Chapter 4.

## Examples

```
.COMPILE PROG,TEST,MAC,MANAGE/COBOL)
```

Compiles PROG (with null extension) with FORTRAN, TEST.MAC with MACRO, and MANAGE (with null extension) with COBOL only if REL files do not exist with later dates. A listing file is generated only for MANAGE. The files generated are PROG.REL, TEST.REL, MANAGE.REL, and MANAGE.LST.

**COMPILE /LIST SIGN,MAC, TABLES/NOLIST, MULTI,ALG )**

Compiles SIGN.MAC with MACRO, TABLES (with null extension) with FORTRAN, and MULTI.ALG with ALGOL. Listing files are generated for SIGN.MAC and MULTI.ALG.

**COMPILE/CREF/COMPILE DIVIDE, SUBTRC, ADD )**

Forces a compilation of the source files although current .REL files exist and generates cross-referenced listing files. The files created are DIVIDE.CRF, DIVIDE.REL, SUBTRC.CRF, SUBTRC.REL, ADD.CRF, and ADD.REL.

**COMPILE /FUDGE;MONITR,REL@LIBALL )**

Compiles the files contained in the command file LIBALL and enters the names of all the REL files generated in a temporary disk file. When the FUDGE command is given, PIP generates the library REL file with name MONITOR.REL. The library is created with the REL files in the same order as they were specified in the command file.

**COMPILE OUTPUT=MTA0:(W,S,M)/L )**

Rewinds the magnetic tape (W), compiles the first file with FORTRAN, produces binary output for the KA10(S), and includes the MACRO coding in the output listing (M). These switches are processor switches (refer to Paragraph 1.5.7). A listing file is generated with the name OUTPUT.LST, along with the file OUTPUT.REL.

## CONTINUE Command

### Function

The CONTINUE command continues the program from the point at which it was interrupted. The program is started at the saved program counter address stored in .JBPC by a HALT command (^C) or a HALT instruction. Refer to DECsystem-10 Monitor Calls for a description of the job data area.

^C followed by CONTINUE has an unexpected action if the program running attempts to rescan (TTCALL 10) the current typed-in line after the CONTINUE. This action returns the CONTINUE instead of the original command. To minimize the probability of this occurring, user programs which rescan the input line should do so at the beginning of the program, to minimize the number of times the user could type a ^C.

### Command Format

CONTINUE

### Characteristics

The CONTINUE command:

Places the terminal in user mode.

Requires core.

Does not require LOGIN.

### Associated Messages

Refer to Chapter 4.

### Example

<u>.RUN LOOP</u> )	Run a program called LOOP in your disk area.
^C	Stop the program.
^C	Check the time of day.
<u>.DAYTIME</u> )	
<u>23-FEB-72 16:33:10</u>	
<u>.CONT</u> )	Continue the program.

## Function

The COPY command transfers files from one standard I/O device to another. The command string can contain one device output specification and any number of input specifications. The equal sign separates the destination (output) side from the source (input) side. This command runs PIP and performs the basic PIP function of transferring files.

## Command Format

```
COPY dev: ^tape id ^file.ext[directory]<nnn> = dev: file.ext  
[directory],...
```

dev: = a physical or logical device name. If the device name is omitted, DSK: is assumed.

^tape id^ = a one-to-six character SIXBIT name to be written as a label for the specified DECTape. The uparrows must be used as delimiters. This argument is optional and is valid only for DECTapes. Refer to the LABEL command description.

file.ext = the name of the file(s) to be used on input or for output. If the output filename is omitted, the input filename is assumed. The files are combined if many input files are being transferred to one output file. If many input files are being transferred to the same number of output files, the files remain as separate files. The wildcard construction is allowed.

[directory] = the disk area in which either the files are to be read or written. When this argument is before the filename, it becomes the default for all succeeding files. If omitted, the user's default disk area is assumed. The user may transfer files to or from another area only if he has access to the area.

<nnn> = the protection code to be given to the output file(s). If omitted, the system standard is assigned, even if the input file already has a non-standard protection associated with it.

Switches can be passed to PIP in the COPY command string by preceding the switch with a slash. When COMPIL interprets the command string, it passes the switches on to PIP.

---

1

This command runs the COMPIL program, which interprets the command before running PIP.

## Characteristics

The COPY command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

## Associated Messages

Refer to Chapter 4.

## Examples

```
_COPY = DTA3; FILNAM,MAC, MANY,CBL, COMMON,ALG )
```

The three files from DTA3 are transferred to the user's disk area with the same filenames.

```
_COPY DTA3; OUTPUT = *.* )
```

All files in the user's disk are transferred to one file on DTA3 with the name OUTPUT.

```
_COPY FILEA,* = DTA1; SOURCE,* )
```

The input files on DTA1 named SOURCE with any extension are transferred to DSK with the filename FILEA and the same extension. The number of output files equal the number of input files.

```
_COPY YOURS,CBL [20, 17] = MINE,CBL )
```

The file MINE.CBL from the user's disk area is transferred to [20,17] disk area with the name YOURS.CBL. The user must have privileges to write in area [20,17].

## CORE Command

### Function

The CORE command types or modifies the amount of core assigned to the user's job. Because programs usually allocate core, the user generally does not need this command. It is included for completeness and is used more frequently in non-swapping systems than in swapping systems.

If the job is locked into core, this command with a nonzero argument cannot be satisfied and therefore gives an erroneous return.

### Command Format

CORE n

n = a decimal number. This argument is optional.

If n is omitted, the monitor types out the amount of core used and does not change the core assignment.

If n = 0, the low and high segments disappear from the virtual addressing space of the job.

If n > 0, n represents the total number of blocks of core to be assigned to the job from this point on.

If n is less than high plus minimum low segment size, n plus high segment size is assumed.

Core arguments can be specified in units of 1024 words or 512 words (a page) by following n with the letter K or P, respectively. For example, 3P represents 3 pages or 1536 words. If K or P is not specified, K (1024 words) is assumed.

On systems with the KA10 processor (DECsystem-1040, 1050, or 1055), 1024 words is the minimum unit of allocation and therefore, all arguments are rounded up to the nearest multiple of 1024 words. For example, 3P on the KA10 is treated the same as 2K.

### Characteristics

The CORE command:

Leaves the terminal in monitor mode.

Does not operate when a device is currently transmitting data.

Associated Messages

Refer to Chapter 4.

Examples

. CORE 5 ↓

. CORE ↓  
5+0/46K CORE  
VIR. CORE LEFT = 274

. CORE 10P ↓  
. CORE  
10+0/93P CORE  
VIR. CORE LEFT = 549P

. CORE ↓  
VIRT. MEM. ASSIGNED 90+6P (CURRENT LIMIT: 400P MAX LIMIT: 512P)  
PHYS. MEM..ASSIGNED 10+6P (CURRENT LIMIT: 40P MAX LIMIT: 512P)  
SWAP SPACE LEFT: 2094P



## CPUNCH Command

### Function

The CPUNCH command is used to place entries into the card punch output queue. This command is equivalent to the following form of the QUEUE command:

```
QUEUE CDP:jobname = list of input specifications
```

### Command Format

```
CPUNCH jobname = list of input specifications
```

jobname = name of the job being entered into the queue. The default is the name of the first file in the request. This may not be the name of the first file appearing in the command string if that file does not yet exist, because the /NEW switch was used. The equal sign can be omitted if the jobname is also omitted.

input specifications = a single file specification or a string of file specifications, separated by commas, for the disk files being processed. A file specification is in the form dev:file.ext [directory].

dev: = any disk file structure to which CDPSPL will have access; the default is DSK:.

file.ext = names of the files. The filename is optional. The default for the first filename is \*, the default for subsequent files is the last filename used. The extension can be omitted; the default is .CDP.

[directory] = a directory to which the user has access; the user's directory is assumed if none is specified.

The user can obtain the listing of entries in the card punch queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., CPUNCH [40,200]). Note that if all arguments to the command are omitted (i.e., only the command name is given), the listing of all entries in the card punch queue for all jobs of all users is output.

The wildcard construction can be used for the input specifications. Switches that aid in constructing the queue entry can also appear as part of the input specifications. These switches are divided into three categories:

1. Queue-operation - Only one of these switches can be placed in the command string because this category defines the type of queue request. The switch used can appear anywhere in the command string.

2. General - Each switch in this category can appear only once in the command string because the category affects the entire request. The switch used can appear anywhere in the command string.
3. File control - Any number of these switches can appear in the command string because this category is specific to individual files with the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the filename, it becomes the default for subsequent files. For example, the command string

CPUNCH FILEA, FILEB/DISP:REN,FILEC

indicates that the DISPOSE switch is only for FILEB.

The command string

CPUNCH/DISP:REN FILEA, FILEB, FILEC

indicates that the DISPOSE switch applies to all three files.

The following switches can be used with the CPUNCH command. Note that if an argument to a switch is omitted, the colon preceding the argument must also be omitted. Otherwise the argument is assumed to be zero and not the default value.

SWITCH	CATEGORY	EXPLANATION
/AFTER:tt	General	Process the request after the specified time; tt is either in the form of hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting AFTER time must be less than the DEADLINE time. If the switch, or the value of the switch, is omitted, no AFTER constraints are assumed.
/BEFORE:t	General	Queue only the files with a creation date before time t where t is in the form dd-mmm-yy hhmm. If the switch, or the value of the switch, is omitted, no BEFORE constraints are assumed.
/BEGIN:n	File Control	Start the output on the nth card. The default is to begin output on the first card.
/COPIES:n	File Control	Repeat the output the specified number of times. N must be less than 64. If more than 63 copies are needed, two separate requests must be made. If this switch is not specified, the default is 1.

SWITCH	CATEGORY	EXPLANATION
/CREATE	Queue Operation	Make a new entry into the card punch output queue. This switch is the default for the queue-operation switches.
/DEADLINE:tt	General	Process the request before the specified time; tt is either in the form hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting DEADLINE time must be greater than the AFTER time. If the switch, or the value of the switch, is omitted, no DEADLINE constraints are assumed.
/DELETE	File Control	Remove the file from the specified directory immediately. (Same as /DISPOSE:RENAME.)
/DISPOSE:DELETE	File Control	Delete the file after it has been punched.
/DISPOSE:PRESERVE	File Control	Do not delete the file after it has been punched. This is the default for all files except files with extensions .LST, .TMP, or .CDP (if protection of .CDP is 0xx).
/DISPOSE:RENAME	File Control	Remove the file from the specified directory immediately. The DELETE option of the DISPOSE switch does not delete the file until it has been punched. Thus, it continues to occupy space in the specified directory. The RENAME option frees the space immediately. This is the default for files with extensions of .LST, .TMP, and, if the protection is 0xx, .CDP.
/F	Queue Operation	List the entries in the card punch queue, but do not update the queues. Therefore, the list may not be an up-to-date listing but the listing will be faster than with /LIST.
/FORMS:a	General	Place the output on the specified form. The argument to the switch must be six alphanumeric characters. The default is that normal forms are used.
/KILL	Queue Operation	Remove the specified entry from the card punch queue. This switch can be used for deleting a previously submitted request as long as the request has not been started by the Spoolers.

SWITCH	CATEGORY	EXPLANATION
/LIMIT:n	General	Limit the output to the specified number of cards.
/LIST	Queue Operation	After updating the queues, list the entries in the card punch queue; if this switch, along with all other switches, is omitted, all entries for all jobs of all users are listed.
/MODIFY	Queue Operation	Alter the specified parameters in the job. This switch requires that the user have access rights to the job. It can be used for altering a previously submitted request as long as the request has not been started by the Spooler.
/NEW	File Control	Accept the request even if the file does not yet exist. An appropriate error message is given if the file does not exist by the time the request is processed by the spooler.
/NULL	General	Do not output an error message if there are no files in the request and do not create a queue entry. This is assumed at KJOB time.
/OKNONE	File Control	Do not output message if no files match the wildcard construction. This is assumed at KJOB time. However, a totally null queue request produces a fatal error message.
/PHYSICAL	File Control	Suppress logical device name assignments for the device specified.
/PRIORITY:n	General	Assign the specified external priority (n = 0 to 62) to the request. The larger the number, the greater priority the job has. The default is 10 if no switch is given and 20 if the switch is specified without a value.
/PROTECT:nnn	General	Assign the protection nnn (octal) to the job. If the switch, or the value of the switch, is omitted, the standard protection is assumed.
/PUNCH:026	File Control	Punch the files in 026 Hollerith code. If a /PUNCH: switch is not given the files are punched according to the data mode specified in the file.

SWITCH	CATEGORY	EXPLANATION
/PUNCH:ASCII	File Control	Punch the files in ASCII card code. If a /PUNCH: switch is not given, the files are punched according to the data mode specified in the file.
/PUNCH:BINARY	File Control	Punch the files in binary card code. If a /PUNCH: switch is not given, the files are punched according to the data mode specified in the file.
/PUNCH:D029	File Control	Punch the files in DEC029 card code. If a /PUNCH: switch is not given, the files are punched according to the data mode specified in the file.
/PUNCH:IMAGE	File Control	Punch the files in image card code. If a /PUNCH: switch is not given, the files are punched according to the data mode specified in the file.
/REMOVE	File Control	Remove the file from the queue. This switch is valid only with the /MODIFY and can be used to remove a previously submitted file as long as CDPSPL has not started processing the request.
/SEQ:n	General	Specify a sequence number to help in identifying a request to be modified or deleted.
/SINCE:t	General	Queue only the files with creation dates after the specified time. t is in the form dd-mmm-yy hhmm.
/START:n	File Control	Begin on the nth line of the file. If the switch, or the value of the switch, is omitted, the spooler starts with the first line.
/STRS	File Control	Search for the file on all file structures in the search list and take each occurrence. The default is to take just the first occurrence.
/UNPRESERVED	General	Output the files only if they are not preserved (i.e., the first digit is 0). This switch avoids redundant punching.

## Characteristics

The CPUNCH command:

Leaves the terminal in monitor mode.

Runs the QUEUE program, thereby destroying the user's core image.

Depends on FTQCOM which is normally absent in the DECsystem-1040.

Does not require LOGIN when only queue listings are desired.

## Associated Messages

Refer to Chapter 4.

## Examples

```
_CPUNCH SYSTAT,MAC/PUNCH:ASCII)
```

Punch the file SYSTAT.MAC in ASCII format.

```
_CPUNCH SYSTAT,REL/PUNCH:BINAR/AFTER:1700)
```

Punch the file SYSTAT.REL in binary format, but do not begin punching it until after 5:00 p.m.

## CREATE Command (1)

### Function

The CREATE command runs LINED (Line Editor for disk) and opens a new file on disk for creation. Refer to the LINED writeup in the DECsystem-10 Software Notebooks.

### Command Format

CREATE file.ext

file.ext = any legal filename and filename extension. The filename is required; the filename extension is optional.

### Characteristics

The Create command:

Places the terminal in user mode.

Runs the LINED program, thereby destroying the user's core image.

Depends on FTCCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Example

```
._CREATE TEST1,F4)
*
```

---

1

This command runs the COMPIL program, which interprets the commands before running LINED.

## CREF Command

### Function

The CREF command runs CREF and lists on the line printer (LPT) any cross-reference listing files generated by previous COMPILE, LOAD, EXECUTE, and DEBUG commands, using the /CREF switch, since the job was initiated. The file containing the names of these CREF-listing files is then deleted so that subsequent CREF commands will not list them again. The output goes either to LPT immediately or to the disk to be spooled later to LPT. When the logical device name LPT is assigned to a device other than the line printer, the CREF files are stored on that device with the same filename and the extension .LST.

### Command Format

CREF

### Characteristics

The CREF command:

Leaves the terminal in monitor mode.

Runs the CREF program, thereby destroying the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Example

**.\_COMPILE/CREF@PROMAC)**

Compile the files contained in the command file PROMAC and produce CREF input compatible cross-reference listing files on the disk.

**.\_CREF**

Process and list the cross-reference listing files produced by the COMPILE command.



**LOAD/C/MAP:NAME@CONALL**

Compile and load the files contained in the command file CONALL. Produce a loader map with the filename NAME and CREF input compatible cross-reference listing files on the disk.

**ASSIGN MTA1 LPT**

Assign the logical name LPT to MTA1.

**CREF**

Store the CREF files on MTA1 to be listed at a later time.

<b>CSTART Command</b> <b>CCONTINUE Command</b>
---

### Function

The CSTART and CCONTINUE commands are identical to the START and CONTINUE commands, respectively, except that the terminal is left in the monitor mode.

### Command Format

CSTART adr  
CCONTINUE

adr = the address at which execution is to begin if other than the location specified within the file (.JBSA). If adr is not specified, the starting address comes from .JBSA. An explicit starting address of 0 may be specified for adr.

To use:

1. Begin the program with the terminal in user mode.
2. Type control information to the program, then type ^C to halt the job with the terminal in monitor mode.
3. Type CCONTINUE to allow job to continue running and leave the terminal in monitor mode.
4. Additional monitor commands can now be entered from the terminal.

### Characteristics

The CSTART and CCONTINUE commands:

Leaves the terminal in monitor mode.

Require core.

Depend on FTATTACH which is normally absent in the DECsystem-1040.

### Restrictions

These commands should not be used when the user program (which is continuing to run) is also requesting input from the terminal. These commands are not available to Batch users.

## Associated Messages

Refer to Chapter 4.

## Example

```
.TYPE LOOP.F4)  
ACCEPT 10,I  
10 FORMAT (I)  
DO 20 J = 1,I  
20 CONTINUE  
END
```

The user types his source program.

```
.EXECUTE LOOP)
```

The user compiles, loads, and executes the program.

```
FORTRAN:LOOP.F4  
LOADING  
LOOP 2K CORE  
EXECUTION  
1000000
```

The user indicates that the program should loop 1000000 times.

```
^C  
^C
```

The user stops the program.

```
.CCONT)
```

The user continues the program but keeps the terminal in monitor mode.

```
.TIME)
```

The user times the program.

```
0.95  
19.62  
KILO-CORE-SEC = 133
```

```
.TIME)
```

The program is still running.

```
1.45  
23.07  
KILO-CORE-SEC = 157
```

```
.SYSTAT)  
PLEASE TYPE ^C FIRST
```

SYSTAT would cause the program to terminate.

```
.TIME)  
0.00  
23.07  
KILO-CORE-SEC = 157
```

The program appears to have finished because the runtime has stopped incrementing. The program will not output until the CONT command is given.

```
^C
```

Return to the monitor.

.CONT)

Continue the program so it can complete its typing.

CPU TIME:5.55 ELAPSED TIME:115.73  
NO EXECUTION ERRORS DETECTED

EXIT

.

## D (deposit) Command

### Function

The D command deposits information in the user's core area (high or low segment). When debugging a sharable program with the D command, the SAVE command should be used rather than the SSAVE command (refer to Appendix D).

When running under a virtual memory system, DAEMON will be called if on a D command the page in question is paged out.

### Command Format

D lh rh adr

lh = the octal value to be deposited in the left half of the location. This argument is required.

rh = the octal value to be deposited in the right half of the location. This argument is required.

adr = the address of the location into which the information is to be deposited. This argument is optional.

If adr is omitted, the data is deposited in the location following the last D adr or in the location of the last E adr (whichever was last).

### Characteristics

The D command:

Leaves the terminal in monitor mode.

Requires core.

### Associated Messages

Refer to Chapter 4.

Example

.D 266000 2616 141      Deposit in location 141.

.E 140  
000140/ 047000 000000      Examine location 140.

.D 47000 1

Since adr is omitted, the deposit is in the location of the last E command.

.E  
000140/ 047000 000001

• The examine is of the location of the previous D command.

## DAYTIME Command

### Function

The DAYTIME command types the date followed by the time of day. The date and time are typed in the following format:

dd-mmm-yy hh:mm:ss

where

dd = day  
mmm = month  
yy = year  
hh = hours  
mm = minutes  
ss = seconds to nearest hundredth.

### Command Format

DAYTIME

### Characteristics

The DAYTIME command

Leaves the terminal in monitor mode.

Does not require LOGIN.

Does not destroy the user's core area.

### Example

```
.DAY )  
11-SEP-70 22:36:34
```

```
.DA )  
15-DEC-71 147:02
```

.

## DCORE Command

### Function

The DCORE command causes the DAEMON program to write a core-image file of the user's core area that includes all accumulators and all relevant job tables. The job can continue to run; i.e., the DCORE command does not destroy the user's core area. The file produced may be later processed by the DUMP program, if the user so desires.

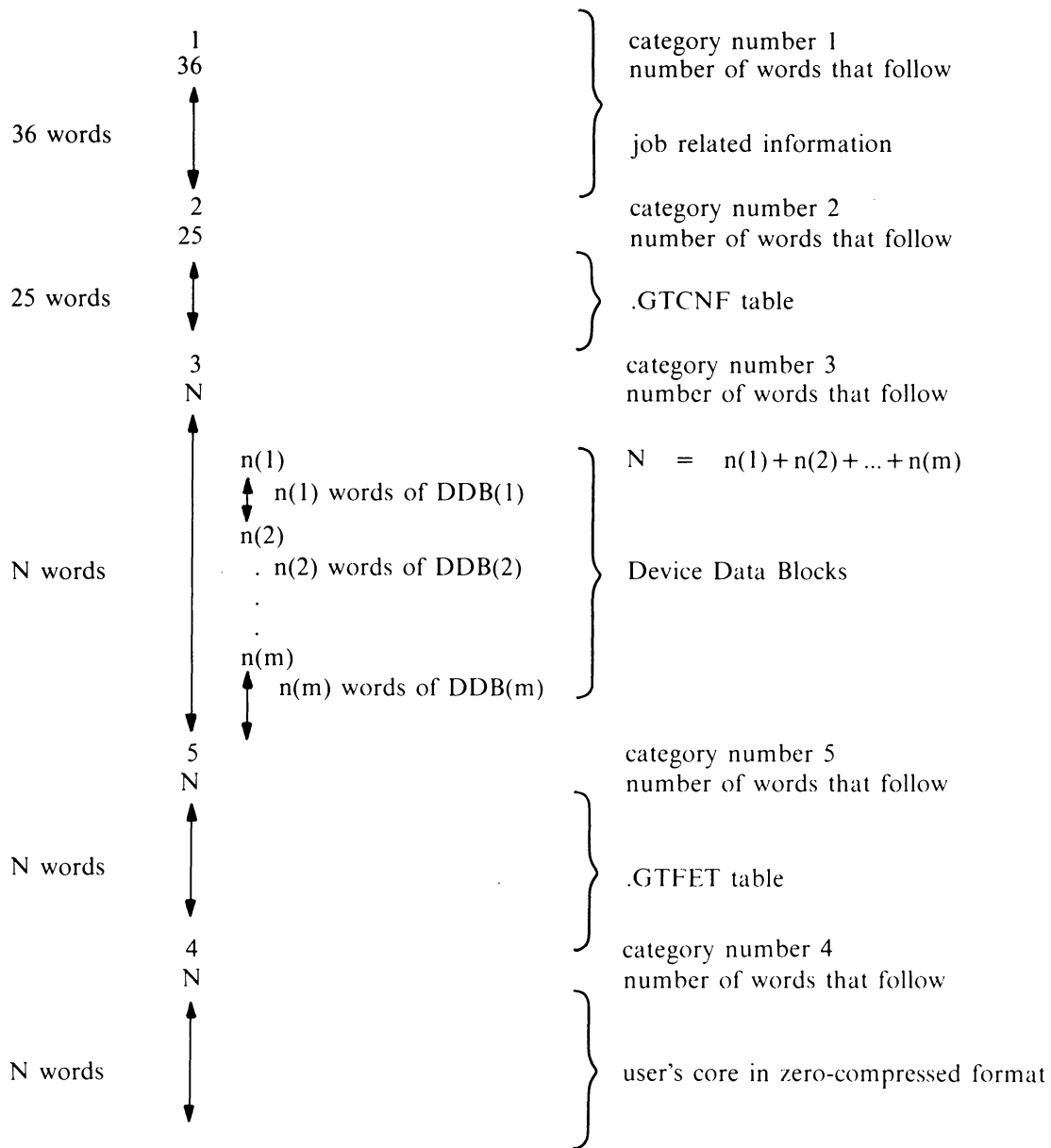
The DAEMON-written file consists of five categories: JOB, CONFIGURATION, DDB, CORE, and FEATURES. Each category has a two-word header, the first word contains the category number and the second word contains the number of data words in the category. The categories are as follows:

Mnemonic	Category Number	Description
JOB	1	Job related information.
CONFIGURATION	2	The Configuration Table (.GTCNF) from the GETTAB UUO.
DDB	3	The device data blocks (DDB) assigned to this job.
CORE	4	The user's core area, both low and high segments, in zero-compressed format (refer to Appendix D).
FEATURES	5	The Feature Table (.GTFET) from the GETTAB UUO.

The third word of each category begins the data for that category.



The DAEMON-written file appears as follows.



Categories may be written in any order. Version 7 of DAEMON writes the categories in the following order: 1,2,3,5,4, but programs should not assume any special order.

Category 1 presently contains the following information, but may expand as more GETTAB entries appear.

Word 1	Version of DAEMON that wrote the file. DATE the file was written in standard system format. TIME in milliseconds that the file was written. JOB NUMBER in left half, HIGH SEG number (or 0) in right half.
--------	---

Word 5	LH is reserved, TTY LINE NUMBER in right half. .GTSTS (job status word) for job. .GTSTS for high segment. .GTPPN (project-programmer number) for job. .GTPPN for high segment. .GTPRG (user program name) for job. .GTPRG for high segment. .GTTIM (total time used) for job. .GTKCT (kilo-core-ticks) for job. .GTPRV (privilege bits) for job. .GTSWP (swapping parameters) for job.
Word 20	.GTSWP for high segment. .GTRCT (disk blocks read) for job. .GTWCT (disk blocks written) for job. .GTTDB (time of day of last disk allocation, number of disk blocks allocated) for job. .GTDEV (device or file structure name) for high segment. .GTNM1 (first half of user's name) for job. .GTNM2 (last half of user's name) for job. .GTCNO (charge number) for job. .GTTMP (TMPCOR pointers) for job. .GTWCH (WATCH bits) for job. .GTSPL (spooling control bits) for job. .GTRTD (real-time status word) for job. .GTLIM (time limit in jiffies) for job. .GTRSP (response counter).
Word 36	.GTSPC (processor status).

Category 2 presently contains the following information, but may expand if more .GTCHF entries are added.

%CNFG0	Name of system in ASCIZ.
%CNFG4	
%CNDT0	Date of system in ASCIZ.
%CNDT1	
%CNTAP	Name of system device in SIXBIT.
%CNTIM	Time of day in jiffies.
%CNDAT	Today's date.
%CNSIZ	Highest location in monitor + 1.
%CNOPR	Name of OPR TTY console.
%CNDEV	LH = beginning of DDB chain.
%CNSGT	LH = -# of high segments, RH = + # of jobs.
%CNTWR	Non zero if system has two register hardware and software.
%CNSTS	LH = feature switches, RH = current state of switches.
%CNSER	Serial number of processor.
%CNNSM	# of nanoseconds per memory cycle.
%CNPTY	PTY parameters for Batch.
%CNFRE	AOBJN word to use bit map in monitor.
%CNLOC	LH = 0, RH = address of free 4-word core block area.

%CNSTB	Link to STB chain for remote Batch.
%CNOPL	OPR LDB Address
%CNTTF	Pointer to TTY free chunks
%CNTTC	LH= number of TTY chunks, RH= address of first
%CNTTN	Number of free chunks
%CNLNS	Pointer to current command TTY
%CNLNP	Pointer to TTY line table
%CNVER	Monitor version
%CNDSC	Pointer to dataset control table
%CNDLS	Last receive int. from DC10
%CNCCI	Last receive int. from 680I
%CNSGT	Last dorm. seg thrown away
%CNPOK	Address of last poked location
%CNPUC	LH= HOB, RH= count of pokes
%CNWHY	Reason for last reload
%CNTIC	Number of ticks per second
%CNPDB	Pointer to PDB Pointer Tables
%CNRTC	Resolution of Run time clock (units/sec)
%CNCHN	LH= PTR to channel D.B list, RH= Unused
%CNLMX	Logmax (max jobs to be logged in)
%CNBMX	BATMAX (Max Batch jobs)
%CNBMN	BATMIN (min jobs reserved for BATCH)
%CNDTM	Internal format date, time
%CNLMN	Number of jobs logged in
%CNBNM	Number of BATCH jobs logged in
%CNYER	Local year
%CNMON	Local month
%CNDAY	Local day of month
%CNHOR	Local hour
%CNMIN	Local minutes
%CNSEC	Local seconds
%CNGMT	Time from GMT in internal format
%CNDDBG	Debugging status word
%CNFRU	Monitor free core used
%CNTCM	Max TTY chunks
%CNCVN	Customer version number
%CNDVN	DEC version
%CNDFC	Number of DF10 data channels
%CNRTD	Number of RT devices
%CNHPQ	Number of HPQ's
%CNLDB	TTY DDB word pointing to LDB
%CNMVO	Max vector offset for PISYS
%CNMIP	Max priority for PISYS
%CNMER	Address of MTA0, offset of MTA err RPT word
%CNETI	User address of exec's AC T1
%CNLSL	length of short DDB
%CNLLD	length of long DDB
%CNLDD	length of disk DDB
%CNEXM	Address in job data area of last e/d command
%CNSTZ	More configuration feature indicators
%CNPIM	Minimum condition in PISYS
%CNPIL	Length of internal PIT's
%CNPIA	Address of JBTPIA

%CNMNT	Monitor type
%CNOCR	First CDR DDB, offset to card count
%CNOCP	Ditto for CCP
%CNPGR	Unit of core allocation

Category 3 contains the device data blocks currently in use for this job. Each DDB is preceded by a word containing the length of the DDB.

Category 4 is a compressed core image of both the high and low segments, i.e., it contains only nonzero words.

Category 5 lists the feature test switches as obtained from the GETTAB table .GTFET.

### Command Format

DCORE dev:name.ext [directory]

dev: = a disk-like device on which the core-image file is to be written. If omitted, DSK is assumed.

name.ext = the name of the file to be written. The default filename is nnnDAE, where nnn is the job number in decimal, and the default extension is .TMP. If a filename is specified, the default extension is DAE.

[directory] = the disk area other than that of the user. If omitted, the user's disk area (the number under which he is logged in) is assumed.

### Characteristics

The DCORE command:

- Leaves the terminal in monitor mode.

- Runs the DAEMON program, without destroying the user's core image.

- Can continue after command.

- Depends on FTDAEM which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

Examples

**DCORE**

The core image file is written on the user's area of the disk with the name nnnDAE.TMP where nnn is the user's job number.

**DCORE DSKB:FILEC**

The core image file is written in the user's area on DSKB with name FILEC.DAE.

**.**

## DDT Command

### Function

The DDT command copies the saved program counter value from .JBPC into .JBOPC and starts the program at an alternate entry point specified in .JBDDT (beginning address of DDT as set by Linking Loader). DDT contains commands to allow the user to start or resume at any desired address. Refer to DECsystem-10 Monitor Calls for a description of the job data area locations.

If the job was executing a UUC when interrupted (i.e., it was in exec mode and not in TTY input wait or SLEEP mode), the monitor sets a status bit (UTRP) and continues the job at the location at which it was interrupted. When the UUC processing is completed, the monitor clears the status bit, sets .JBOPC to the address following the UUC, and then traps to the DDT address found in .JBDDT. If the job is in exec mode and in TTY input wait or SLEEP mode, the trap to the DDT address occurs immediately and .JBOPC contains the address of the UUC. If the job is in user mode, the trap also occurs immediately. Therefore, it is always possible to continue the interrupted program after trapping to DDT by executing a JRSTF @ .JBOPC.

For additional information on the DDT program, refer to the DDT Programmer's Reference Manual in the DECsystem-10 Software Notebooks.

### Command Format

DDT

### Characteristics

The DDT command:

Places the terminal in user mode.

Requires core.

Requires the user to have a job number.

### Associated Messages

Refer to Chapter 4.

Examples

```
.TYPE LOOP,MAC)  
LOOP: JRST LOOP  
END LOOP
```

Type an undebugged program.

```
.DEBUG LOOP  
MACRO: MAIN  
LOADING
```

Assemble load the program with DDT.

```
LOOP 3K CORE  
DDT EXECUTION
```

```
*Z  
.SAVE)  
JOB SAVED
```

Save the program.

```
.START
```

Start the program.

```
*C  
*C
```

Stop it.

```
.DDT)
```

Enter DDT.

```
LOOP/JRST LOOP CALLI12  
JRSTF @,JBOPC&X
```

Fix the program.

```
EXIT
```

```
.
```

# DEASSIGN Command

## Function

The DEASSIGN command returns one or more devices currently ASSIGNED to the user's job back to the monitor pool of available devices and clears any logical names. Restricted devices are returned to the restricted pool, and unrestricted devices to the unrestricted pool. Note that an INITed device is not returned to the monitor pool unless a RELEASE UUC is done, only the logical name is cleared. Therefore, this command is provided for programs that are not terminating or programs that are being debugged. The command, applied to DECTapes, clears the copy of the directory currently in core, forcing the next directory reference to read a new copy from the tape. (Refer to DECsystem-10 Monitor Calls for further details.

## Command Format

DEASSIGN dev

dev = either the logical or physical device name. This argument is optional. If it is not specified, all devices assigned to the user's job, except the job's controlling terminal, are deassigned, and the logical name of the controlling terminal is cleared.

## Characteristics

The DEASSIGN command:

Leaves the terminal in monitor mode.

## Associated Messages

Refer to Chapter 4.

## Examples

```
DEASSIGN LPT1 )  
:
```

The line printer is returned to the monitor's pool of available resources.

```
DEASSIGN )  
:
```

All devices assigned to the job are returned.



**Function**

The DEBUG command translates the specified source files if necessary (function of the COMPILE command), loads the REL files generated (function of the LOAD command), and prepares for debugging. A system debugging program is loaded first, followed by the user's program with local symbols. Upon completion of loading, control is transferred to the debugging program. The debugging program used depends upon the first file in the command string. If this file is a COBOL source file, COBDDT (the COBOL debugging program) is used. If the first file is any file other than a COBOL source file, DDT (the Dynamic Debugging Technique) is loaded.<sup>(2)</sup> When the first file is a previously translated program (i.e., with extension .REL) and COBDDT is desired as the debugging program, the /COBOL switch must appear in the command string; for example,

DEBUG FILEA.REL/COBOL

Refer to the DDT documentation for a description of the DDT commands, and to the COBOL documentation for a description of COBDDT.

In general, the use of the DEBUG command requires more core to execute a program than the EXECUTE command requires. The extra space is occupied by the system debugging program and additional debugging information, such as the local symbols.

Each time a COMPILE, LOAD, EXECUTE, or DEBUG command is executed, the command with its arguments is remembered in a temporary file on disk, or in core if the monitor has the TMPCOR feature. Therefore, issuing one of these commands without any arguments causes the arguments saved in the temporary file to be reused (refer to Paragraph 1.5).

The DEBUG command accepts several command constructions: the @ construction (indirect commands), the + construction, the = construction, and the <> construction. Refer to Paragraph 1.5 for a complete description of each of these constructions.

**Command Format****DEBUG list**

list = a single file specification, or a string of file specifications separated by commas. A file specification consists of a device name, a filename with or without an extension, and a directory name (refer to Paragraph 1.4.2.4).

---

1

This command runs the COMPIL program, which interprets the command before running the appropriate language translator, the LOADER, and the appropriate debugger.

2

MANTIS will be recognized as a debugging program only if the appropriate assembly switch is set. However, this switch setting is not supported. If the switch is set and the first file in the command string is a FORTRAN source file, MANTIS (the FORTRAN debugging program) is loaded.

The following switches can be used to modify the command string. These switches can be temporary or permanent unless stated otherwise (refer to Paragraph 1.5.5).

/ALGOL	Compile the file with ALGOL. Assumed for files with the extension of .ALG.
/BIN	Generate a binary file for each file compiled. The filename of the binary file follows the standard conventions for determining the filename of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .REL. This is the default action.
/BLISS(1)	Compile the file with BLIS10. Assumed for files with the extension of .B10 and .BLI.
/COBOL	Compile the file with COBOL. Assumed for files with the extension of .CBL.
/COMPILE	Force a compilation on this file even if a binary file exists with a newer date and time than the source file. This switch is used to obtain an extra compilation (e.g., in order to obtain a listing of the compilation) since normally compilation is not performed if the binary file is newer than the source file.
/CREF	Produce a cross-reference listing file on the disk for each file compiled for later processing by the CREF program. The filename for the listing file follows the standard convention for determining the name of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .CRF. The file can then be listed with the CREF command. However, with COBOL files, the cross-reference listing is always appended to the listing file. No additional command need be given to obtain the listing.
/DDT	Load DDT regardless of the extension of the first file in the command string. This is a permanent switch and applies to all subsequent files.
/FOROTS	Load the file with FOROTS (the new FORTRAN object-time system).
/FORSE	Load the file with FORSE (the old FORTRAN object-time system).
/FORTRAN	Compile the file with a FORTRAN compiler. Assumed for files with the extension of .F4 and .FOR and all files with nonrecognizable translator extensions if FORTRAN is the

---

<sup>1</sup> BLIS10 will be recognized as translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

standard translator. This switch is needed if the file has a non-recognizable translator extension and FORTRAN is not the standard translator or is not the current default (e.g., DEBUG/COBOL TEST1, TEST2, TEST3/ FORTRAN).

**/FUDGE** Create a disk file containing the names of the .REL files produced by the command string. When the FUDGE command is given, PIP reads this file in order to generate a library REL file (refer to the FUDGE2 program description). Arguments to this switch are:

`/FUDGE:dev:file.ext [proj, prog]`

`dev`: - the device on which to write the file. If the device is omitted, `DSK`: is assumed.

`file.ext` - the name of the library file. The filename is required. If the extension is omitted, it is assumed to be `.REL`.

`[proj, prog]` - the directory in which to place the file. The user's directory is assumed if none is given.

This switch is permanent in the sense that it pertains to all REL files generated by the command string.

**/F10** Use the FORTRAN-10 compiler when compiling the associated FORTRAN file. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.

**/F40** Use the F40 compiler when compiling the associated FORTRAN file. This is the current default action. This should be used as permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.

**/KA10**  
**/K110** Designate the machine on which the program will execute once it has been loaded. These switches are needed for FORTRAN-10 programs since the compiler generates different code for the KA10 and K110 processors. The `/KA10` switch is the current default.

**/LIBRARY** Load the files in library search mode. This mode causes a program file in a special library file to be loaded only if one or more of its declared entry symbols satisfies an undefined global request in the source file. The system libraries are always searched. Refer to the `LOADER` and `LINK-10` documentation.

**/LINK** Cause the file to be loaded by the `LINK-10` linking loader. If used, this switch should be placed before any file specifications since the `COMPIL` program may have to generate load-control switches.

/LIST	Generate a disk listing file, for each file compiled. The filename for the listing file follows the standard conventions for determining the name of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .LST. These files can be listed later with the LIST command. If the line printer is being spooled for this job, the listing files are written on device LPT and are automatically spooled at LOGOUT time. Unless this switch is specified, listing files are not generated.
/LMAP	Produce a loader map during the loading process (same action as /MAP) containing the local symbols.
/LOADER	Cause the file to be loaded by the LOADER program. This is the current default action.
/MACRO	Assemble the file with MACRO. Assumed for files with extensions of .MAC.
/MACX11(1)	Assemble the file with MACX11. Assumed for files with the extension .P11.
/MANTIS(2)	Compile the file with the MANTIS debugging information. This switch affects FORTRAN programs only.
/MAP	Produce a loader map during the loading process. When this switch is encountered, a loader map is requested from the loader. After the library search of the system libraries, the map is written in the user's disk area with either the filename specified by the user (e.g., /MAP:file) or the default filename MAP.MAP if loading is performed by LOADER or nnnLNK.MAP if performed by LINK-10. This switch is an exception to the permanent switch rule in that it causes only one map to be produced even though it may appear as a permanent switch.
/NEW	Run the appropriate language translator from the experimental system library (device NEW:) area [1,5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:. See the following NOTE.
/NOBIN	Do not generate binary files. Unless this switch is given, binary files are generated. This switch, when combined with the /LIST or /CREF switch, is useful when compiling programs solely for the purpose of generating listings.

---

<sup>1</sup> MACX11 (the PDP-11 assembler for the PDP-10) will be recognized as a language translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

<sup>2</sup>

MANTIS will be recognized as a debugging program only if the appropriate assembly switch is set. However, this switch setting is not supported. If the switch is set and the first file in the command string is a FORTRAN source file, MANTIS (the FORTRAN debugging program) is loaded instead of DDT.

/NOCOMPILE	Complement the /COMPILE switch by not forcing a compilation on a source file whose date is not as recent as the date on the binary file. Note that this switch is not the same as the /REL switch, which turns off all compilation, even if the source file is newer than the REL file. /NOCOMPILE is the default action.
/NOLIST	Do not generate listing files. This is the default action.
/NOMANTIS(1)	Compile the file without the MANTIS debugging information. This switch affects FORTRAN programs only.
/NOSEARCH	Loads all routines of the file whether the routines are referenced or not. Since this is the default action, this switch is used only to turn off library search mode (/LIBRARY). This switch is not equivalent to the /P switch of the LOADER, which does not search any libraries. The /NOSEARCH default is to search the system libraries.
/OLD	Run the appropriate language translator from the system library of old programs (device OLD:) which resides on the disk area [1,3]. If the translator does not exist on device OLD:, try to obtain it from device SYS:. See the following NOTE.
/REL	Use the existing REL files although newer source files may be present.
/SEARCH	The action is identical to that of the /LIBRARY switch.
/SELF	Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for an individual who keeps a private copy of a translator in his own disk area. System programmers occasionally keep experimental versions of standard translators in their disk areas in order to test new features. See the following NOTE.
/SNOBOL(2)	Compile the file with SNOBOL. Assumed for files with extensions of .SNO.
/SYS	Run the appropriate processor from the system library (device SYS:) area of [1,4]. This is the default action.

---

<sup>1</sup> MANTIS will be recognized as a debugging program only if the appropriate assembly switch is set. However, this switch setting is not supported. If the switch is set and the first file in the command string is a FORTRAN source file, MANTIS (the FORTRAN debugging program) is loaded instead of DDT.

<sup>2</sup> SNOBOL will be recognized as a language translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

#### NOTE

Once a processor has been specified from a particular area (e.g., /SELF), it cannot be called from a different area within the same command string; i.e., the following is illegal:

```
.DEBUG MAIN.MAC/SELF,PART1.MAC/OLD
```

However, the following is valid:

```
.COMPIL MAIN.MAC/SELF  
.COMPIL PART1.MAC/OLD  
.DEBUG /REL MAIN,PART1
```

#### Characteristics

The DEBUG command:

Places the terminal in user mode.

Runs the appropriate processor, the LOADER, and the debugger, thereby destroying the user's original core image.

#### Associated Messages

Refer to Chapter 4.

#### Examples

```
.DEBUG/L FILEA,FILEB,FILEC/N,FILED)
```

Generate listings for FILEA, FILEB,  
and FILED

```
.DEBUG TEST)  
MACRO: TEST  
LOADING
```

```
LOADER 2K CORE  
DDT EXECUTION
```

```
./ BLT 15,0(16)
```

## DELETE Command (1)

### Function

The DELETE command deletes one or more files from disk or DECtape. Note that the protection codes normally assigned to files prevent one user from deleting files in another user's directory.

### Command Format

#### DELETE list

list = a single file specification or a string of file specifications separated by commas. The full wildcard construction (\* and ?) can be used. Refer to Paragraph 1.4.2.4.

If a device or file structure name is specified, it remains in effect until changed or until the end of command string is reached. When a directory name is before the filename, it becomes the default for all succeeding files. A directory name after a filename applies only to that file.

### Characteristics

#### The DELETE command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

---

1

This command runs the COMPIL program, which interprets the command before running PIP.

Examples

```
.DEL *,MAC)
FILES DELETED:
T1,MAC
T2,MAC
T3,MAC
14 BLOCKS FREED
```

```
.DEL TEST1,MAC)
FILES DELETED:
TEST1,MAC
3 BLOCKS FREED
```

.

```
.DEL TEST??.F4)
FILES DELETED:
TEST,F4
TESTS,F4
TEST03,F4
TEST2,F4
23 BLOCKS FREED
```



## DETACH Command

### Function

The DETACH command disconnects the terminal from the user's job without affecting the status of the job. The user terminal is now free to control another job, either by initiating a new job or attaching to a currently running detached job.

### Command Format

DETACH

### Characteristics

The DETACH command:

Detaches the terminal.

Depends on FTATTACH which is normally absent in the DECsystem-1040.

### Restrictions

This command is not available to Batch users.

### Associated Messages

Refer to Chapter 4.

### Example

```
.DETACH )  
FROM JOB 1
```

.

## DIRECT Command

### Function

The DIRECT command lists the directory entries specified by the argument list. The standard output consists of the following columns: filename, filename extension, length in blocks written, protection, creation date, version number, structure name, and directory name. Refer to the DIRECT specification in the DECsystem-10 Software Notebooks for additional information on the DIRECT program.

### Command Format

DIRECT output specification = list of input specifications

list = A single file specification, or a string of file specifications separated by commas or plus signs. The devices used on input can be DSK:, DTA:, MTA:, and TMP: (TMPCOR). If the device is a magnetic tape, the tape is rewound before and after the listing operation and analyzed to determine if it is a FAILSAFE tape. The default input specification is DSK:\*.\*, and the user's directories in all file structures defined by the job's search list are listed. Generally, a device name, an extension, or a directory name that precedes the filename becomes the default for all succeeding files in the list.

The full wildcard construction (\* and/or ?) can be used on input specifications. However, when a wildcard designation is used, the DIRECT program limits its search for the file to certain directories. When the user gives a wildcard designation for a filename or extension, the program only searches the specified directory or the user's default directory. No additional directories, such as LIB or SYS, which may be in the user's default path are searched. (Refer to the SETSRC program description in this manual and to the PATH, UWO description in DECsystem-10 Monitor Calls. If the user has a wildcard designation for a directory, only the directories implied by the wildcard construction are searched. Again, no additional directories are searched. When the user gives a file specification without any wildcard designations, the DIRECT program uses the normal procedures for locating the file. That is, the user's path, LIB, NEW, and SYS as enabled by the user via the SETSRC program or the PATH, UWO are searched.

output specification = This argument (and the equal sign) is optional. If the entire output specification is omitted, the default is TTY:. If an output filename is given, the default device is DSK:. If an output filename is not given, and one is needed, the filename is generated from the time of day as hhmmss. The default output extension is .DIR. The wildcard construction cannot be used in the output specification.

The following switches may be used in the command string. In general, non-complementary switches can be used together in the same command string. Switches that precede the filename become the default for all succeeding files.

Switch names can be truncated as long as the resulting abbreviation is unique. Spaces are not permitted with a switch (i.e., between the slash at the beginning of the switch and the argument at the end).

- `/ACCESS:n` Update the access date to the current date for any file of `n` blocks or less accessed (i.e., listed) by the DIRECT program. Since some installations delete files that have not been recently accessed, this switch allows the user to prevent such deletion by updating the date. `n` is interpreted as a decimal number and refers to the number of blocks actually written in the file unless the `/ALLOC` switch is also used. If `/ACCESS` is omitted, the date is not changed. If `/ACCESS` is specified but `:n` is omitted, `n = 5` is assumed.
- `/ALLOC` List the allocated length of the file instead of the written length. Space on a structure is sometimes allocated in units of more than one block for efficiency. Therefore, the number of blocks allocated to a file may be greater than the number of blocks actually written. The allocated length is used by LOGOUT in checking quotas. The total allocated length of all files is the same as the length output by the QUOLST program under the USED column. Complement of `/WRITTEN`. (Disk and magnetic tape only.)
- `/BEFORE:date:time` List only those files created before the specified date and time. Default is +infinity. Refer to Paragraph 1.4.2.5.
- `/BLOCKS` Output the length of the file in blocks instead of words. Complement of `/WORDS`. This is the default.
- `/CHECKSUM` Compute and print an 18-bit checksum for each file. This checksum is computed by rotating the result left one bit before adding each word. Complement of `/NOCHECKSUM`. (Disk and magnetic tape only.)
- `/DENSITY:n` Use the specified density when reading a magnetic tape. `N` is 200.556, or 800 bpi. The default is installation dependent and is modified by the SET DENSITY command.
- `/DETAIL` Print all available information about a file except for zero values (refer to the extended LOOKUP block in DECsystem-10 Monitor Calls for a list of the values which are printed). The protection and data mode are also listed, even if they are zero. The author is not listed if it is the same as the owner of the directory. Numbers followed by a period are decimal numbers. All other numbers are octal. (Disk and magnetic tape only.)

/EOTS	Stop at the logical end of tape (two consecutive tape marks) when reading a magnetic tape. Complement of /NOEOTS. This is the default.
/FAST	List short form of directory (i.e., filename, extension, structure name, and directory name). Equivalent to /F. Complement of /NORMAL and /SLOW.
/FILES:n	Stop after n tape marks (files) when reading a magnetic tape. If /FILES is specified but :n is omitted, n = 1 is assumed. Note that the logical EOT will also stop unless /NOEOTS is specified.
/HELP	Help text which indicates some of the switches available and how to use them. Equivalent to /H.
/HELP:S	List all switches (S) without their explanations. An asterisk prefixes those switches which have a single-letter abbreviation.
/LIST	List the output on device LPT:. Equivalent to /L.
/MARKS	Indicate each tape mark, including the final tape mark, and UFD when reading a magnetic tape. Complement of /NOMARKS.
/NOCHECKSUM	Do not compute and print the checksum. Complement of /CHECKSUM. This is the default.
/NODETAIL	Do not list the words in the LOOKUP block. Complement of /DETAIL. This is the default.
/NOEOTS	Do not stop at the logical end of tape when reading a magnetic tape. Complement of /EOTS.
/NOMARKS	Do not indicate each tape mark and UFD when reading a magnetic tape. Complement of /MARKS. This is the default.
/NORMAL	Output the normal directory listing. This listing includes the filename, extension, length in blocks written, protection, creation date, structure name, non-zero version numbers, and directory name. Complement of /FAST and /SLOW. This is the default. The switch is used to override a /FAST or /SLOW in the user's option file.
/NOREWINDS	Do not rewind the tape before and after reading a magnetic tape. Complement of /REWINDS.
/NOSORT	Do not produce a file suitable for sorting. Complement of /SORT. This is the default.

<code>/NOSUMMARY</code>	Do not use summary mode (i.e., output more than just the summary line). Complement of <code>/SUMMARY</code> . This is the default.
<code>/NOTITLE</code>	Do not output page headers. Complement of <code>/TITLES</code> . This is the default for output to the terminal.
<code>/NOUNITS</code>	Do not list the name of the actual disk unit; instead, just list the structure name. Complement of <code>/UNITS</code> . This is the default.
<code>/OKNONE</code>	Suppress the error message if no files match the wildcard construction.
<code>/OPTION:name</code>	Read the user's option file ( <code>DSK:SWITCH.INI[,]/PHYSICAL</code> ) to determine the user's specified switch defaults for <code>DIRECT</code> . The name appearing as the value of the switch is the pointer to the line to read in the file. For example, if the file contains the line  <p style="text-align: center;"><code>DIRECT:ALL/DETAIL</code></p> then the user references this line by typing the command  <p style="text-align: center;"><code>DIRECT/OPTION:ALL</code></p> Refer to Paragraph 1.4.3 for additional information.
<code>/PARITY:ODD</code> <code>/PARITY:EVEN</code>	Specify the parity to be used when reading a magnetic tape. The default is <code>ODD</code> .
<code>/PHYSICAL</code>	Ignore logical names used for device names (refer to the <code>ASSIGN</code> command for a description of logical names).
<code>/PROTECTION:nnn</code>	Give the output file the protection <code>nnn</code> (octal).
<code>/REWINDS</code>	Rewind the magnetic tape before and after reading it. Complement of <code>/NOREWINDS</code> . This is the default.
<code>/RUN:file spec</code>	Run the specified program when this command is finished.
<code>/RUNOFFSET:n</code>	Run the program specified with <code>/RUN</code> with an offset of <code>n</code> . If the switch is omitted, the default is 0; if the switch is given without a value, the default is 1.
<code>/SINCE:date:time</code>	List only those files created after the specified date and time. The default is January 1, 1964. Refer to Paragraph 1.4.2.5.

/SLOW	Output a full listing that includes the filename, extension, length in blocks written, protection, creation date and time, access date, structure name, and directory name. Equivalent to /S. Complement of /FAST and /NORMAL. (Disk and magnetic tape only.)
/SORT	List the file structure name, if there is more than one or if the files are on magnetic tape, and directory name, if a wildcard directory is given, on each line instead of only on the first line in which they change. Multiple spaces are output instead of TABs. Project-programmer numbers include leading zeroes; the date is in ANSI format (e.g., 19721009 for Oct.9,1972). This switch is used to prepare a file to be sorted by the SORT program (refer to the DECsystem-10 COBOL Programmer's Reference Manual). Complement of /NOSORT.
/SUMMARY	Output only the summary line which indicates the total number of blocks and files. Note a /FAST /SUMMARY lists a /FAST listing followed by the summary.
/TITLES	Cause a heading to be output on each page consisting of a label for each column, date, time, and page number. Standard output to the line printer has this heading. Complement of /NOTITLE.
/UNITS	List the name of the actual disk unit on which the files are stored instead of the file structure name. Complement of /NOUNITS.
/WIDTH:n	Output several entries on a single line to make the output n columns wide. For example, if /FAST is specified for output to a terminal, four filenames appear per line. The default for n is 64 columns.
/WORDS	Output the length of the file in words instead of blocks. Complement of /BLOCKS.
/WRITTEN	Return the written length of the file rather than the allocated length. Complement of /ALLOC. This is the default.

## Characteristics

The DIRECT command:

Leaves the terminal in monitor mode.

Runs the DIRECT program, thereby destroying the user's core image.

Depends on FTCCCLX which is normally absent in the DECsystem-1040. In this case, run the DIRECT program with the R command and input the appropriate command strings following the asterisk response from DIRECT.

Examples

- `.DIR DTA3)` Lists all files on DTA3.
- `.DIR *,MAC)` Lists all files with extension .MAC in all file structures in the user's job search list.
- `.DIR TEST.F4[27,60])` Lists the directory entry for file TEST.F4 in user area 27, 60.

The ordinary default directory.

```
.DIRECT
FORMA3      5 <055> 17-MAY-74   DSKC:  [27,4072]
FORMA2      3 <055> 17-MAY-74
TOFCMI     12 <055> 30-MAY-74
PLAN       16 <055> 30-MAY-74
PLAN3      8 <055> 31-MAY-74
PLAN3      8 <055> 31-MAY-74
PLAN2     11 <055> 31-MAY-74
SCOPLA      5 <055>  3-JUN-74
TOTAL OF 80 BLOCKS IN 8 FILES ON DSKC: [27,4072]
```

The same directory reporting blocks allocated instead of written.

```
.DIRECT/ALLOC
FORMA3      10 <055> 17-MAY-74   DSKC:  [27,4072]
FORMA2      5 <055> 17-MAY-74
TOFCMI     15 <055> 30-MAY-74
PLAN       20 <055> 30-MAY-74
PLAN3      10 <055> 31-MAY-74
PLAN3      25 <055> 31-MAY-74
PLAN2     15 <055> 31-MAY-74
SCOPLA      10 <055>  3-JUN-74
TOTAL OF 110 BLOCKS IN 8 FILES ON DSKC: [27,4072]
```

The full detail directory for the same files.

```
.DIRECT/DETAIL
DSKC: FORMA3.[27,4072]
ACCESS DATE: 17-MAY-74
CREATION TIME, DATE:  9:34 17-MAY-74
ACCESS PROTECTION: 055
MODE: 1
WORDS WRITTEN: 610.
BLOCKS ALLOCATED: 10.
DATA BLOCK IN DIRECTORY: 58470.
INTERNAL CREATION DATE, TIME: 17-MAY-74  9:34:17
RIP BLOCK NUMBER: 31255.
```

.  
. .  
. .

DSK00:FORMA2.[27,4072]  
ACCESS DATE: 17-MAY-74  
CREATION TIME, DATE: 14:19 17-MAY-74  
ACCESS PROTECTION: 055  
MODE: 1  
WORDS WRITTEN: 336.  
BLOCKS ALLOCATED: 5.  
DATA BLOCK IN DIRECTORY: 68470.  
INTERNAL CREATION DATE, TIME: 17-MAY-74 14:19:17  
RIP BLOCK NUMBER: 40100.

DSK00:TOFCMI.[27,4072]  
ACCESS DATE: 30-MAY-74  
CREATION TIME, DATE: 12:06 30-MAY-74  
ACCESS PROTECTION: 055  
MODE: 0  
WORDS WRITTEN: 1475.  
BLOCKS ALLOCATED: 15.  
DATA BLOCK IN DIRECTORY: 68470.  
INTERNAL CREATION DATE, TIME: 30-MAY-74 12:06:42  
RIP BLOCK NUMBER: 58010.

**\_DIRECT [40, 1]** Lists the directory entries for user with project number 40 and the user's programmer number.

Example of an incorrect specification in the /BEFORE switch.

**\_DIR SYSN:\*,SAV/SINCE:15-OCT-1972/BEFORE:NOVEMBER)**  
**? MISSING DAY IN DATE/TIME**

Example of using both /BEFORE and /SINCE to specify a range of dates.

**\_DIR SYSN:\*,SAV/SINCE:15-OCT-1972/BEFORE:1-NOV)**

CRASH	SAV	PROTECTION	FAILURE	DSKN:	[1,4]
P1402B	SAV	438	<155>	17-OCT-72	(50402
R1524H	SAV	529	<155>	17-OCT-72	(50524)
FRI525	SAV	209	<155>	18-OCT-72	(34(70)
RI525C	SAV	528	<155>	20-OCT-72	(50525)
FRI526	SAV	212	<155>	25-OCT-72	34(70)
RI526C	SAV	535	<155>	30-OCT-72	(50526)



RI526D	SAV	535	<155>	30-OCT-72	(5026)
RI527A	SAV	533	<155>	1-NOV-72	(50527)

TOTAL OF 3519 BLOCKS IN 8 FILES AND 1 LOOKUP ERROR ON  
 DSKN: [1,4]

Example of user [10,251] with LIB of [7,5062]

\_DIR \*,MAC[,,\*,\*],\*.TXT[,,\*,\*],LIB:\*,MCO )

DUMP	MAC	200	<055>	13-AUG-72	DSKB:	[10,251]
FILUUD	MAC	309	<055>	20-OCT-72		
G	MAC	16	<055>	20-OCT-72		
R	MAC	16	<055>	20-OCT-72		
T	MAC	1	<055>	20-OCT-72		
REMDLX	MAC	62	<055>	20-OCT-72		
MICRO	MAC	26	<055>	30-OCT-72		

TOTAL OF 630 BLOCKS IN 7 FILES ON DSKB: [10,251]

FILUUD	MAC	309	<055>	20-OCT-72	[10,251,A]
MONITR	TXT	1	<055>	2-NOV-72	[10,251,A,B]
506	MCO	97	<100>	1-NOV-72	BLKA: [7,5062]

GRAND TOTAL OF 1037 BLOCKS IN 10 FILES

## DISMOUNT Command

### Function

The DISMOUNT command allows a user to return devices to the monitor pool of available resources and to remove a file structure from his search list. Restricted devices are returned to the restricted pool and unrestricted devices to the unrestricted pool. The command applied to non-file structures is identical to the DEASSIGN command if the user waits for completion of the operator action. If the user does not wait for completion (e.g., he types a control-C after the message OPERATOR NOTIFIED), the device is not deassigned, but the request to the operator is still queued for the purpose of removing the media. The user must then issue the DEASSIGN command to release the device. This command applied to file structures enforces logged-out quotas (if necessary), allows physical removal of disk packs (if there are no other users of the pack), and removes the file structure name from the job's search list.

The UMOUNT program runs privileged in the user's core area when the DISMOUNT command is typed. This program scans the user's command string, checks its validity, and performs as much of the requested action as possible. The UMOUNT program can complete all actions requested by the DISMOUNT command except for the action of physically removing packs, tapes, or cards. When operator action is required, the UMOUNT program writes a command file on [3,3] disk area and notifies the OMOUNT program (running on the operator's terminal) to perform the action. When the operator action has been completed, OMOUNT deletes the command file and notifies UMOUNT (if UMOUNT is waiting) to inform the user of completion.

To insure validity of any tape error analysis, MOUNT/DISMOUNT should be used to acquire and release magtape units. This mechanism provides the basis for all media-related error reporting.

### Command Format

DISMOUNT dev:switches

dev: = any previously ASSIGNed or MOUNTed device or file structure name. This argument is required.

switches = the following switches are optional and only enough characters to make the switch unique are required.

/CHECK	Check and list pending requests.
/HELP	Type this list.
/PAUSE	Notify the user before requesting operator action. The user can then abort the command if desired.

/REMOVE

Notify operator to physically remove disk packs, tape, or cards. A file structure is removed from the system only if no other users are using it. A request to remove the pack is queued to the operator and the message WAITING...is typed to the user. If the user does not want to wait for confirmation of the operator action, he may type control-C. This switch must be specified to notify the operator to remove the pack, even if no other jobs are using it.

### Characteristics

The DISMOUNT command:

Places the terminal in user mode.

Runs the UMount program, thereby destroying the user's core image.

Depends on FTCLX and FTMOUN which are normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Examples

```
.DISMOUNT DSKA: )  
DSKA DISMOUNTED
```

The user dismounts the file structure DSKA. This does not require an operator action.

```
.DISMOUNT DTA4: /R )  
OPERATOR NOTIFIED
```

The user asks the operator to deassign DTA4 and remove the tape.

```
WAITING...
```

The command is waiting for completion of the operator action.

```
^C
```

The user does not wish to wait for confirmation of removal.

```
.DISMOUNT/CHECK )  
NONE PENDING  
0 COMMANDS IN QUEUE  
:
```

The user checks for completion and determines that his request is finished.

## DSK Command

### Function

The DSK command types disk usage for the combined structures of the job, since the last DSK command, followed by the total disk usage since the job was initialized (logged in). Disk usage is typed in the following format:

$$\begin{aligned}RD,WT &= I,J \\RD,WT &= M,N\end{aligned}$$

where I and J are the incremental number of 128-word blocks read and written since the last DSK command, and M and N are the total number of 128-word blocks read and written since the job was initialized.

#### NOTE

I and J are kept modulo 4096. If automatic READ or WRITE print outs have been enabled using the SET WATCH command, I and J are usually zero, since the SET WATCH output also resets these values.

### Command Format

#### DSK job

job = the job number of the job for which the disk usage is desired. This argument is optional.

If job is omitted, the job to which the terminal is attached is assumed.

If job is supplied (whether the job of this user or another user) the incremental quantities are not reset to zero.

### Characteristics

The DSK command:

Leaves the terminal in monitor mode.

### Associated Messages

Refer to Chapter 4.

Example

.DSK )  
RD,WT=12,0  
RD,WT=475,243

!

## DTCOPY Program

### Function

The DTCOPY program is a DECTape copy routine that allows the user to

1. Copy the entire contents of an input DECTape to an output DECTape.
2. Zero all blocks on an output DECTape and clear the directory.
3. Perform a word-by-word comparison of two DECTapes.
4. Load a bootstrap loader and write it in blocks 0, 1, and 2 of the output DECTape.

### Command Format

.R DTCOPY

\*output DTA: = input DTA:/switches

/switches = one or more of the following switches. Switches are preceded by a slash or enclosed in parentheses and can appear anywhere in the command string.

- /C Copy all blocks from the input DECTape to the output DECTape.
- /G Do not restart the program after a parity error. Output an error message and continue the program.
- /H Type the available switches and their meanings.
- /L Load the bootstrap loader into a core buffer. COPY expects the loader to be on logical device PTR in the file named BSLDR.REL. Note that COPY must be SAVED if the loader is to be preserved with the COPY core image.
- /N Suppress the directory listing.
- /T Write the bootstrap loader in blocks 0, 1, and 2 of the output DECTape. This switch accepts, as input from the terminal, a core bank or offset. The loader is offset and then written on the tape.  
  
core bank = nnnK (16K to 256K)  
offset = 1000 to 777600 octal
- /V Verify the similarities of the two DECTapes by performing a word-by-word comparison and typing on the terminal the number of discrepancies discovered.

/Z Zero all blocks of the output DECTape and clear the directory.

/6 Look for the directory in PDP-6 format (i.e., in block one instead of block 144).

If no switches are specified, /C (copy) and /V (verify) are assumed by default. Note that upon completion, the directory in core may not agree with the directory of the output DECTape. The output DECTape should be reassigned to guarantee that the directory in core is up-to-date.

## Characteristics

The R DTCOPY command:

Places the terminal in user mode.

Runs the COPY program, thereby destroying the user's core image.

## Associated Messages

Refer to Chapter 4.

## Examples

<code>␣R DTCOPY␣</code>	Run DTCOPY
<code>*DTA7␣ ␣ DTA3␣␣</code>	Copy the contents of DTA3 to DTA7 and determine if the two DECTapes are the same (default condition). If the DECTapes disagree, the number of discrepancies is typed on the terminal.
<code>*DTA2␣/Z ␣ ␣</code>	Zero all blocks and clear the directory on DTA2.
<code>*C</code>	Return to monitor mode.
<code>␣ASSIGN DSKIPTR␣␣</code>	The bootstrap loader must be on logical device PTR.
<code>␣RENAME BSLDR,REL ␣ DTBOOT,REL␣␣</code>	COPY expects the bootstrap loader to be named BSLDR.
<code>␣R DTCOPY␣</code>	Run DTCOPY
<code>*L ␣</code>	Load the bootstrap loader into a core buffer.

**\*^C** Return to monitor mode.

**.\_SAVE DSKICOPY )** Save DTCOPY so that the bootstrap loader is preserved with the DTCOPY core image.

**.\_START )** Start the COPY program.

**\*DTA5I/T = )** Write the bootstrap loader in blocks 0, 1, and 2 of DTA5.

**TYPE CORE BANK AND OFFSET FOR DTBOOT**  
Respond with size of core bank or offset.

**64K )** Size of core bank (64K core bank = 177000 offset, top of core -1000)

**\*^C** Return to monitor mode.



## DUMP Command

### Function

The DUMP command calls the DAEMON program to write a core image file (function of the DCORE command) and then invokes the DUMP program to analyze the file written and to provide printable output. The core image file is named nnnDAE.TMP where nnn is the user's job number. This file is described in detail in the DCORE command description.

### Command Formats

1. DUMP /command/command/command...
2. DUMP @ dev:file.ext [directory]
3. DUMP

The commands that appear in the DUMP command string are passed to the DUMP program and therefore are described in the DUMP program description. A DUMP command using a command file can also specify these commands. A DUMP command without any arguments prints a short dump of the user's core area via the command file QUIKDM.CCL which resides on device SYS:.

### Characteristics

The DUMP command:

Leaves the terminal in monitor mode.

Runs the DAEMON and DUMP programs, thereby destroying the user's core image.

Depends on FTDAEM which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

## Examples

```
.DUMP/OUT:TTY:/MODE:ASCII,SIXBIT/WIDTH:7,10/JUST:L,R- )  
/RIGHTMA:26/D [3000 & 3004
```

This command string writes a core image file named nnnDAE.TMP and invokes the DUMP program to perform the output. The output goes to the terminal and the modes used on output are ASCII and SIXBIT. The ASCII field is 7 characters long, left justified and the SIXBIT field is 10 characters long, right justified. The right margin of the output is 26 characters. The dump consists of the contents of word 3000 to word 3004. The hyphen is used to continue the command string onto the next line.

**Function**

The DUMP program provides printable dumps of arbitrary data files in modes and forms specified by the user. The DUMP program accepts any data file as input and produces an ASCII file suitable for listing by PIP, the output spoolers, or other listing programs. For example, the DUMP program takes core image files prepared by the DAEMON program or SAVED files produced by the monitor. For a description of the DAEMON-written file, refer to the DCORE command. Complete documentation on the DUMP program is contained in the DUMP Specification and in the DUMP Reference Manual is the DECsystem-10 Software Notebooks.

**Command Formats**

1. R DUMP  
/command
2. R DUMP  
/@ dev:file.ext [directory]

**NOTE**

DUMP indicates its readiness by typing a slash (/) instead of an asterisk.

The commands with their arguments are as follows. Lines can be continued by typing a hyphen followed by a carriage return.

Command	Argument	Meaning
ADDRESS	ON or OFF	Specifies if the address is to be dumped along with its contents. The default is ON.
ALL		Dumps the entire file. If the file is a DAEMON core image file, the entire category is dumped.
APPEND		Appends the output to the output file. The existing output file is not overwritten. This command is the default; its complement is SUPERSEDE.
AUTOFORMAT	ON or OFF	Attempts to format output with line feeds, form feeds, and titles, if ON. If OFF, the user is responsible for all formatting. The default is ON.

Command	Argument	Meaning
CATEGORY	mnemonic for name of category Can be JOB, CONFIGURATION, DDB, CORE, or FEATURES.	Selects the category of the DAEMON dump file to be used. Addressing begins with 0 at the beginning of each category. The default category is CORE. If the input file is not a DAEMON file, this switch has no effect. Note that the DUMP program does not display the category header nor does it allow the user to read past the end of one category into the next category.
CLOSE		Closes the output file. After this command is given, another OUT command must be given before the next command which does any output or else a fatal error message will result.
DUMP or D	dump descriptor, dump descriptor....	Dumps the specified bytes in the current modes.
EJECT		Starts a new page in the output file.
EXIT		Closes all files and returns control to the monitor (^ Z has the same effect).
HELP or H		Types the help text on the user's terminal. To list the names of all the switches, the user types HELP: SWITCHES.
INPUT or I	file descriptor	Specifies the input file. The defaults are: DSK:nnnDAE.TMP where nnn is the job number; the user's directory. If the filename is specified, it determines the extension from the set .TMP, .DAE, .SHR, .SAV, .HGH, .LOW, .XPN, and .DMP in that order. If an extension is specified with no filename, the extension determines the filename.
IRADIX	decimal number	Specifies radix for numbers for input. This command uses decimal to compute the argument. The default is 10 for decimal. The argument must be numeric. If the argument is 0 or is missing, the input radix is set back to its default value.

Command	Argument	Meaning
JUSTIFY	LEFT, CENTER, or RIGHT	Specifies the justification of the output in the output field. If the output overflows the output field, the entire output appears; it is not truncated. This switch is used in a one-to-one relationship with the MODE and WIDTH commands. If there are more MODE commands, an argument of LEFT is used. If there are more JUSTIFY commands, they are ignored.
LEFTMARGIN	expression	Sets the left margin of the output file. The default is 0.
LINEPAGE	expression	Specifies the number of lines per output page. This counts all lines including blank lines and titles. The default is 50.
MODES or M	ALL, ASCII, DECIMAL, NULL, NUMERIC, OCTAL, RADIX50, SIXBIT, SOCTAL, or SYMBOLIC	<p>Selects the type of output.</p> <p>All dumps in all modes. ASCII dumps the word as a single right justified character if bits 0-28 are zero or as 5 left justified ASCII characters if bits 0-28 are nonzero. Non printing characters print as a space. DECIMAL dumps as a signed decimal number. NULL declares that nothing is to be dumped. NUMERIC dumps as a signed number in the current output radix. OCTAL dumps as half-words separated by a comma (default) and takes 13 positions. RADIX50 dumps in RADIX50. SIXBIT dumps as one right justified SIXBIT character if bits 0-29 are zero, or 6 SIXBIT characters if bits 0-29 are nonzero. SOCTAL dumps as signed octal and suppresses leading zeroes. SYMBOLIC dumps as a symbolic instruction.</p> <p>Any mode specification can appear more than once in the command string. The output is in the same order as the MODE list.</p>
NUMPAGE	expression	Specifies that pages are to be numbered. If expression is 0, page numbering is turned off. If expression is not 0, page numbering begins at page = [expression]. If command is omitted, numbering starts at the first page.

Command	Argument	Meaning
OPTION	:name	<p>Reads the user's option file (DSK: SWITCH.INI[,] /PHYSICAL) to determine the user's specified switch defaults for this program. The option name appearing as the value of the switch is the pointer to the line to read in the file. For example, if the file contains the line</p> <p style="text-align: center;">DUMP:OCT/IRADIX:8/ORADIX:8</p> <p>then the user references this line by typing to DUMP</p> <p style="text-align: center;">/OPTION:OCT</p> <p>If SWITCH.INI has a line of the form</p> <p style="text-align: center;">DUMP/switch/switch</p> <p>then the specified switches override the normal DUMP defaults. For example, if the user does not want the address dumped, he can place the line</p> <p style="text-align: center;">DUMP/ADDRESS:OFF</p> <p>in the file. If he wants the addresses dumped for a particular dump, he must override the file by giving the command</p> <p style="text-align: center;">/ADDRESS:ON</p> <p>to DUMP</p>
ORADIX	decimal number	<p>Specifies radix for numbers for output. The default is 10 for decimal. If number is 0, the standard is used. The argument to this command is decimal and must not be an expression.</p>
OUTPUT or O	file descriptor	<p>Specifies the output file. The defaults are: LPT:, the filename of the input file; the extension .LSD; the user's directory. If a filename is given, the device DSK: is assumed.</p>
RIGHTMARGIN	expression	<p>Sets the right margin of the output file. A field may overflow the right margin if it will not fit between the left and right margins. If ADDRESS is ON, the new</p>

Command	Argument	Meaning
		line will have an address typed. If a page overflow occurs, a title line may also be printed.
RUN	program	Runs the specified program. This command is equivalent to the R monitor command.
SUPERSEDE		Specifies that the output is to supersede an existing file of the same name, if there is one. The complement of this command is APPEND, which is the default.
SYFILE	file descriptor	Specifies the file to take symbols from if XTRACT command is specified. Defaults are: DSK:, the filename of the input file; one of the saved file extensions; the user's directory area.
TDUMP or T	dump descriptor, dump descriptor,...	Dumps specified bytes to both output file and TTY.
TITLE	string of characters	Specifies a title to be included in the subsequent page headings. If no argument is specified, titling is turned off.  After this command, an EJECT command should be given to skip to a new page.
TYPE(1)	DAE,DAT,HGH, LOW, SAV, SHR, XPN	Specifies the format of the input or symbol file. DAE specifies the file is generated by DAEMON. DAT specifies file is a data file (i.e., no special format; therefore, no special processing is done). HGH specifies file is in .HGH format. LOW specifies file is in .LOW format. SAV specifies file is in .SAV format. SHR specifies file is in .SHR format. XPN specifies file is in .XPN format.
WIDTH	expression	Selects the width of each output mode (see the MODE and JUSTIFY commands). If a MODE command is specified without a corresponding WIDTH,

---

If TYPE is not specified, the extension of the input file is used to determine the type of file being produced. If the extension is not one recognized in the TYPE command, TYPE DAE is assumed.

Command	Argument	Meaning
		the byte is dumped in exactly the number of positions required followed by 3 blanks. If a WIDTH command is specified, no free blanks are output. If a MODE specification overflows its WIDTH specification, the entire output is given without justification. If expression is omitted, justification is turned off.
XTRACT		Uses the file specified in the last SYFILE command as a core image and extracts the symbol table.

An expression is an octal or decimal number, arithmetic operations using expressions (+, -, \*, /, ^, and ^ grouped with parentheses), contents operator ([, \, and @), or symbols. A symbol is a string of SIXBIT characters, or program: symbol, where program defines the program containing symbol. The following symbols are built into the DUMP program:

- \$ (the last byte typed out),
- . (the address of the last word dumped), and
- % (the last expression evaluated).

A dump descriptor is any of the following:

1. A text string enclosed in single quotes which contains alphanumeric characters and special patterns. The following patterns represent nongraphic characters and are replaced in the output by the action represented.

- <EL> - end line, <CR-LF>
- <VT> - vertical tab
- <FF> - form feed
- <AL> - altmode
- <HT> - horizontal tab
- <letter> - control character
- <letter> - lower case character

To override special patterns, a double quote indicates that the next character is to be represented as is, without including it as part of a special pattern.

2. A byte descriptor describing the byte in the input file to be dumped. The format is:

WORD <POS, SIZE>

where

WORD = the address of the word desired.



POS = the position of the byte within the word. It specifies the bit number of the leftmost bit in the byte.

SIZE = the number of bits in the byte. It may be any size and can cross word or block boundaries.

3. A byte descriptor limit specifying everything from the first byte descriptor to the second. The format is as follows:

<FROM byte descriptor> & <TO byte descriptor>

A file descriptor consists of a device name, a filename with an extension, and a directory name.

### Characteristics

The R DUMP command:

Places the terminal in user mode.

Is used with disk monitors only.

Runs the DUMP program, thereby destroying the user's core image.

### Associated Messages

Refer to Chapter 4.

### Example

```
.DCORE )           ;MAKE A CORE IMAGE FILE
.R DUMP )
/XTRACT           ;GET THE SYMBOL TABLE

643 SYMBOLS EXTRACTED
/MODE SYM,OCT,DEC ;DUMP IN SYMBOLIC, OCTAL AND DECIMAL
/WIDTH 30,15,15  ;ALLOW 30 PRINT POSITIONS FOR SYMBOLIC COL.,
/              ; 15 FOR OCTAL AND DECIMAL,
/JUSTIFY L,R,R,R,R ;JUSTIFY INSTRUCTIONS LEFT EVERYTHING ELSE RIGHT
/OUT TTY:        ;SEND OUTPUT TO TTY:
/RIGHTMARGIN 70 ;SET LINE WIDTH TO 70
/IRADIX 8        ;CHANGE THE INPUT RADIX TO OCTAL
/DUMP [140&150] ;DUMP THE CONTENTS OF 140 THRU 150

140/   JSR   SAVE           264000,001454   24159191852
141/   PUSHJ P,REMOVB      260040,001773   23630709755
142/   MOVEI T,XEC1       201240,001447   17356030759
143/   HRRM  T,PROCo      542240,001634   -21164457060
144/   PUSHJ P,CRF        260050,002774   23630710268
```

145/	TLZ	SCH	621000,000010	-14898167800
146/	MOVE	T,DC8+5	200240,003504	17221814084
147/	BLT	T,ODF	251240,000012	22724739082
150/	SETZM	PRNC	402000,003613	-34091300981
/EXIT				

,ALL DONE

## E (examine) Command

### Function

The E command examines a core location in the user's area (high or low segment).

When running under a virtual memory system, DAEMON will be called on execution of the E command, if the page in question is paged out.

### Command Format

E adr

adr is required the first time the E or D command is used. If adr is specified, the contents of the location are typed out in half-word octal mode.

If adr is not specified, the contents of the location following the previously specified E adr or the location of the previous D adr (whichever was last) are typed out.

### Characteristics

The E command:

Leaves the terminal in monitor mode.

Requires core.

### Associated Messages

Refer to Chapter 4.

### Example

```
.E 140 )  
000140/ 264000 002616 .E  
000141/ 000000 000000 .E  
000142/ 000000 000000 .
```

## EDIT Command (1)

### Function

The EDIT command opens an already existing line sequence-numbered file on disk for editing with LINED (Line Editor for disk). Refer to the LINED writeup in the DECsystem-10 Software Notebooks.

### Command Format

EDIT file.ext

file.ext = a filename and filename extension of an existing file. This argument is optional if a CREATE or EDIT command has been given since the initialization of the job, because the arguments of the EDIT-class commands are remembered in temporary files on the disk or in core if the monitor has the TMPCOR feature.

### Characteristics

The EDIT command:

Places the terminal in user mode.

Runs the LINED program, thereby destroying the user's core image.

Depends on FTCCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Example

```
EDIT TEST.F4 )  
|  
|
```

---

1

This command runs the COMPIL program, which interprets the command before running LINED.

## EOF Command (1)

### Function

The EOF command writes an end of file mark on the specified magnetic tape. This command is equivalent to the following PIP command string:

MTAn: (MF) \_

### Command Format

EOF MTAn:

More than one magnetic tape can be specified in the command string by separating the tape specifications with commas.

### Characteristics

The EOF command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Examples

```
._EOF MT31,2)
```

```
!
```

---

<sup>1</sup> This command runs the COMPIL program, which interprets the command before running the PIP program.

## EXECUTE Command (1)

### Function

The EXECUTE command translates the specific source files if necessary (function of COMPILE command), loads the REL files generated into a core image (function of LOAD command), and begins execution of the program. The language translator used is determined from the source file extensions or from switches in the command string (refer to the COMPILE command). If a REL file already exists with a newer date than that of the source file, compilation is not performed (unless requested explicitly via a switch).

This command is equivalent to a LOAD and START sequence of commands.

Each time a COMPILE, LOAD, EXECUTE, or DEBUG command is executed, the command with its arguments is remembered in a temporary file on disk, or in core if the monitor has the TMPCOR feature. Therefore, issuing one of these commands without any arguments causes the arguments saved in the temporary file to be reused (refer to Paragraph 1.5).

The EXECUTE command accepts several command constructions: the @ construction (indirect commands), the + construction, the = construction, and the <> construction. Refer to Paragraph 1.5 for a complete description of each of these constructions.

### Command Format

#### EXECUTE list

list = a single file specification, or a string of file specifications separated by commas. A file specification consists of a device name, a filename with or without an extension, and a directory name (refer to Paragraph 1.4.2.4).

The following switches can be used to modify the command string. These switches can be temporary or permanent switches unless otherwise stated (refer to Paragraph 1.5.5).

/ALGOL                      Compile the file with ALGOL. Assumed for files with the extension of .ALG.

/BIN                              Generate a binary file for each file compiled. The filename for the binary file follows the standard conventions for determining the name of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .REL. This is the default action.

---

<sup>1</sup> This command runs the COMPIL program, which interprets the command before running the appropriate language translator for the LOADER.

/BLISS(1)	Compile the file with BLISS10. Assumed for files with the extension of .B10 or .BLI.
/COBOL	Compile the file with COBOL. Assumed for files with the extension of .CBL.
/COMPILE	Force a compilation on this file even if a binary file exists with a newer date and time than the source file. This switch is used to obtain an extra compilation (e.g., in order to obtain a listing of the compilation) since normally compilation is not performed if the binary file is newer than the source file.
/CREF	Produce a cross-reference listing file on the disk for each file compiled for later processing by the CREF program. The filename for the listing file follows the standard conventions for determining the name of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .CRF. The files can then be listed with the CREF command. However, with COBOL files, the cross-referenced listing is appended to the listing file. No additional command need be given to obtain the listing.
/FOROTS	Load the file with FOROTS (the new FORTRAN object-time system).
/FORSE	Load the file with FORSE (the old FORTRAN object-time system).
/FORTRAN	Compile the file with a FORTRAN compiler. Assumed for files with the extension of .F4 and .FOR and all files with nonrecognizable translator extensions if FORTRAN is the standard translator. This switch is needed if the file has a nonrecognizable translator extension and FORTRAN is not the standard translator or is not the current default (e.g., EXE/ALGOL.FIL1.FIL2.FIL3/FORTRAN).
/FUDGE	Create a disk file containing the names of the .REL files produced by the command string. When the FUDGE command is given, PIP reads this file in order to generate a library REL file (refer to the FUDGE2 program description). Arguments to this switch are:

/FUDGE:dev:file.ext [proj, prog]

---

1

BLISS10 will be recognized as a language translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

dev: - the device on which to write the file. If the device is omitted, DSK: is assumed.

file.ext - the name of the library file. The filename is required. If the extension is omitted, it is assumed to be .REL.

[proj, prog] - the directory in which to place the file. The user's directory is assumed if none is given.

This switch is permanent in the sense that it pertains to all REL files generated by the command string.

/F10 Use the FORTRAN-10 compiler when compiling the associated FORTRAN file. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.

/F40 Use the F40 compiler when compiling the associated FORTRAN file. This is the current default action. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.

/KA10 Designate the machine on which the  
/KI10 program will execute once it has been loaded. These switches are needed for FORTRAN-10 programs since the compiler generates different code for the KA10 and KI10 processors. The /KA10 switch is the current default.

/LIBRARY Load the files in library search mode. This mode causes a program file in a special library file to be loaded only if one or more of its declared entry symbols satisfies an undefined global request in the source file. The system libraries are always searched. Refer to the LOADER and LINK-10 documentation.

/LINK Cause the file to be loaded by the LINK-10 linking loader. If used, this switch should be placed before any file specifications since the COMPIL program may have to generate load-control switches.

/LIST Generate a disk listing file, for each file compiled. The filename for the listing file follows the standard conventions for determining the name of the output file (refer to



	Paragraphs 1.5.2 and 1.5.3). The extension is .LST. These files can be listed later with the LIST command. If the line printer is being spooled for this job, the listing files are written on device LPT and are automatically spooled at LOGOUT time. Unless this switch is specified, listing files are not generated.
/LMAP	Produce a loader map during the loading process (same action as /MAP) containing the local symbols.
/LOADER	Cause the file to be loaded by the LOADER. This is the current default action.
/MACRO	Assemble the file with MACRO. Assumed for files with extensions of .MAC.
/MACX11(1)	Assemble the file with MACX11. Assumed for files with extensions of .P11.
/MANTIS(2)	Compile the file with the MANTIS debugging information. This switch affects FORTRAN files only.
/MAP	Produce loader maps during the loading process. When this switch is encountered, a loader map is requested from the loader. After the library search of the system libraries, the map is written in the user's disk area with either the filename specified by the user(e.g., /MAP:file) or the default filename MAP.MAP if loading is performed by LOADER or nnnLNK.MAP if performed by LINK-10. This switch is an exception to the permanent switch rule in that it causes only one map to be produced even though it appears as a permanent switch.
/NEW	Run the appropriate language translator from the experimental system library (device NEW:) area [1.5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:. Refer to the following NOTE.

---

<sup>1</sup> MACX11, the PDP-11 assembler for the PDP-10, will be recognized as a translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

<sup>2</sup> MANTIS will be recognized as a debugging program only if the appropriate assembly switch is set. However, this switch setting is not supported.

/NOBIN	Do not generate binary files. Unless this switch is given, binary files are generated. This switch, when combined with the /CREF or /LIST switch, is useful when compiling programs solely for the purpose of generating listings.
/NOCOMPILE	Complement the /COMPILE switch by not forcing a compilation on a source file whose date is not as recent as the date on the binary file. Note that this switch is not the same as the /REL switch, which turns off all compilation, even if the source file is newer than the REL file. /NOCOMPILE is the default action.
/NOLIST	Do not generate listing files. This is the default action.
/NOMANTIS(1)	Compile the file without the MANTIS debugging information. This switch affects FORTRAN programs only.
/NOSEARCH	Loads all routines of the file whether the routines are referenced or not. Since this is the default action, this switch is used only to turn off library search mode (/LIBRARY). This is not equivalent to the /P LOADER switch, which does not search any libraries; the /NOSEARCH switch scans the system libraries.
/OLD	Run the appropriate language translator from the system library of old programs (device OLD:) which resides on the disk area [1,3]. If the translator does not exist on device OLD:, try to obtain it from device SYS:. Refer to the following NOTE.
/REL	Use the existing REL files although newer source files may be present.
/SEARCH	The action is identical to that of the /LIBRARY switch.
/SELF	Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for an individual who keeps a private copy of a translator in his own disk area. System

---

1  
MANTIS will be recognized as a debugging program only if the appropriate assembly switch is set. However, this switch setting is not supported.

programmers occasionally keep experimental versions of standard translators in their disk areas in order to test new features. Refer to the NOTE below.

/SNOBOL(1)	Compile the file with SNOBOL. Assumed for files with an extension of .SNO.
/SYS	Run the appropriate language translator from the system library (device SYS:) area [1,4]. This is the default action.

#### NOTE

Once a language translator has been specified from a particular area within the same command string, i.e., the following is illegal:

```
.EXECUTE PARTA.F40/NEW, PARTB.F40/OLD
```

However, the following is valid:

```
.COMPIL PARTA.F40/NEW  
.COMPIL PARTB.F40/OLD  
.EXECUTE/REL PARTA, PARTB
```

#### Characteristics

The EXECUTE command:

Places the terminal in user mode.

Runs the appropriate language translator and linking-loader, thereby destroying the user's original core image.

Starts the execution of the compiled and loaded program.

#### Associated Messages

Refer to Chapter 4.

---

<sup>1</sup> SNOBOL will be recognized as a language translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

Examples

EXECUTE TEST)  
MACRO: TEST  
LOADING

LOADER 2K CORE  
EXECUTION

**Function**

The FAILSAFE program is used by operators and users alike to save disk files on magnetic tape and to later place all or some of these saved files back onto the disk. Although this program is normally used by the system operator, the console user can execute a subset of the FAILSAFE operations for his own disk area. He can save his disk area on his own magnetic tape and later restore all his files or a subset of his files back to his area. Or he may need to restore from the system failsafe tape one of his files that he unintentionally deleted from his area. The user can also use the program for moving his files from one system to another. Complete documentation on the FAILSAFE program is contained in the FAILSAFE specification in the DECsystem-10 Software Notebooks.

**WARNING**

Crucial to FAILSAFE'S operation is the fact that the magnetic tape used must be assigned the logical name FAILSA.

**Command Format**

.R FAILSAFE  
\*/switch

FAILSAFE switches with their explanations are as follows. A carriage return is used to terminate all switches.

<b>SWITCH</b>	<b>EXPLANATION</b>
/A	Advances the magnetic tape one physical file.
/B	Backspaces the magnetic tape one physical file.
/C	Causes FAILSAFE to continue. <ol style="list-style-type: none"><li>1. On disk-to-tape transfers, if the end of tape is reached before all disk files have been saved, FAILSAFE requests that the user mount a new tape and type /C after the tape has been mounted to direct FAILSAFE to continue.</li><li>2. On tape-to-disk transfers, after FAILSAFE has typed out the number of the mounted tape and its creation time and date, the user types /C if this information is correct and he wishes all files on the tape to be restored on disk. This feature aids in preventing accidental restoration of files from the wrong tape.</li></ol>

SWITCH	EXPLANATION
/D	Transfers control of DDT if it is loaded.
/E	<p>Sets creation date and time word so old files can be eliminated by not being transferred in either direction. The format is as follows:</p> <p style="text-align: center;">*/E mm/dd/yy, tttt&lt;carriage return&gt;</p> <p>where mm is the numerical month, dd is the day of the month, yy is the last two digits of the year, and tttt is the time expressed in 24-hour time. If this switch is not specified, Jan. 1, 1964, 0000 hrs, is used.</p>
/F	<p>Sets access date word. This switch is used to transfer, in either direction, only recently accessed files or to save on tape and then delete from disk not recently accessed files (/K before /S or /U). The format is as follows:</p> <p style="text-align: center;">*/F mm/dd/yy &lt;carriage return&gt;</p> <p>where mm is the numerical month, dd is the day of the month, and yy is the last two digits of the year. If this switch is not specified, Jan. 1, 1964 is used.</p>
/G	<p>Enables the user to save and restore files from a user's area other than his own. This switch sets the source project-programmer number to the specified number. The format is as follows:</p> <p style="text-align: center;">*/G mmm.nnn&lt;carriage return&gt;</p> <p>This changes the single-user project-programmer number switch from that of the FAILSAFE user to mmm, nnn. The new value is retained until the next /G switch. If this switch is not given, the project-programmer number of the user running the job is used.</p>
/H	Prints a summary of FAILSAFE operating procedures on the user's terminal.
/I	Sets the magnetic tape density to the installation standard.
/J	Looks for the next trailer record (which marks the end of a save set) and points to the beginning of the next save set on the tape or to the logical end of tape, if there is no save set. In the first case, the tape is positioned in such a way that a tape read command will execute properly or a tape write

**SWITCH****EXPLANATION**

command will overwrite the existing save set. If there is no save set on the tape, the tape is positioned between the two end of file marks designating the logical end of tape, so that a succeeding tape write (/S or /U) will execute properly. Use the /A switch to position the tape after MAGRIM and/or SAVE and SSAVE header files.

/K Deletes not-recently accessed files from disk and copies them to a magnetic tape. (This is known as the skim operation because, in one pass, the disk is being skimmed of certain files and at the same time these files are placed on the tape). Used in conjunction with /S or /U. Note that certain project-programmer numbers may be exempted from this process.

/L Types on the user's terminal, a directory of all the user's files on the tape. FAILSAFE checks the current user's project-programmer number and uses it to find the correct area of the tape. Only the filenames and extensions are typed. This switch is similar to the /F switch in PIP.

/M Initiates multiple saves. Saves are taken every SLPMIN minutes where SLPMIN is an assembly parameter initially set to 60 (decimal) minutes.

/N Inhibits checking of creation dates of tape and disk files of same name when restoring from tape. Therefore, FAILSAFE restores a file from tape over the file of the same name on the disk without regard to creation date. This switch remains in effect until the line of input is terminated by a carriage return-line feed.

/O Enables the user to save and restore files to a user's area other than his own. This switch sets the destination project-programmer number to the specified number. The format is as follows:

\*/O xxx.yyy <carriage return>

This changes the single-user project-programmer number switch from that of the FAILSAFE user to xxx.yyy. The new value is retained until the next /O switch. If this switch is not given, the project-programmer number of the user running the job is used.

**SWITCH****EXPLANATION**

/P	Prints a directory of all files on the tape on logical device LST. The directory is listed by project-programmer number and includes the number of blocks allocated per file, a running total for each area, and creation time/access date information. If the logical device LST is assigned to the disk or a DEctape, the filename FAILSA.DIR is given to the directory file.
/Q	Disables the detailed progress reports given by FAILSAFE on a single-user restore. Complement of /T. This is assumed unless changed by a /T switch.
/R	Restores all files from the tape to the disk. This switch does not accept an argument because it transfers the entire tape. The user must be logged in under [1,2] to use this feature.
/S	Saves disk files on the magnetic tape. Up to 16 arguments can follow this switch, separated by spaces and/or tabs. These arguments are names of file structures or disk devices from which files are to be taken. If no arguments are specified, the entire disk is saved; in this case the user must be logged in under [1,2].
/T	Tells the user, on a single-user restore, the user area that has been found on the tape, the file that has been selected for transfer to the disk, when the transfer is begun, and when the transfer is completed. An example of this progress report is shown below:  LOCATED ON TAPE DSKC 27,235 FOUND ON TAPE DSKC:FILENAME,EXT <>  There may be a possible pause between the time the angle brackets are typed at the beginning and the end of transfer. The /T switch takes effect on the file for which it appears and ends when a /Q switch is given.
/U	Saves, on the magnetic tape, the disk files in the UFD of the user last named in a /G switch, or if no /G switch has been given, of the user currently logged in. Up to 16 arguments, separated by spaces and/or tabs, can follow this switch. These arguments are names of file structures or disk devices from which the files are to be taken, and when used, only those structures mentioned that have files for the specified user, are saved. If no arguments are given, all the user's files on all the structures are saved. The user is informed if no files are saved from a specified structure.



SWITCH	EXPLANATION
/V	Generates a request to lock the job in core, or resets the request. Note that locking may not be accomplished even though the request is given. FAILSAFE acknowledges if the lock is successful and does not acknowledge if the lock is unsuccessful. This switch is a toggle switch in the sense that the first occurrence sets the request, the second resets the request, and so forth. The user can consult SYSTAT or SYSDPY to determine the state of the switch.
/W	Rewinds the magnetic tape with the logical name FAILSA to load point.
/X	Extracts the project-programmer numbers from the tape and reproduces output similar to the original TTY output generated by FAILSAFE during a save. The output contains the names and the project-programmer numbers on the tape. It is placed on logical device LST, and if LST is assigned to the disk or DECTape, the filename FAILSA.DIR is given to the output.
/Y	Used to debug new features.
/Z	Used to debug new features.
/2	Sets the magnetic tape density to 200 bpi.
/5	Sets the magnetic tape density to 556 bpi.
/8	Sets the magnetic tape density to 800 bpi.

The user can be selective as to the files he saves or restores. Files are specified either explicitly by giving the filename and extension or implicitly by giving the wildcard construction.

#### Characteristics

The R FAILSAFE command:

Places the terminal in user mode.

Runs the FAILSAFE program, thereby destroying the user's core image.

Examples:

1. Saving a user's disk area

```
.MOUNT MTA; FAILSA /VID:FAILSAFE-TAPE
OPERATOR NOTIFIED
WAITING...
FAILSA (MTAO) MOUNTED

.R FAILSA

REWINDS ARE NOT AUTOMATIC,
FAILSAFE VERSION 7700,13; FOR HELP, READ SYS:FAILSA,
HLP, OR
TYPE /H
*/W
*/U
10:47:50          03 AUG 72
TAPE # 01 @ 9 CH 200 BPI
NO FILES SAVED FOR 20,1547    DSKA
DSKC    20,1547
NO FILES SAVED FOR 20,1547    DSKB

$SAVE COMPLETED WITH TAPE # 01
10:47:57          03 AUG 72

*
```

2. Restoring a user's disk area

```
.MOUNT MTA; FAILSA /VID:FAILSAFE-TAPE
OPERATOR NOTIFIED
WAITING...
FAILSA (MTAO) MOUNTED

.R FAILSA

REWINDS ARE NOT AUTOMATIC,
FAILSAFE VERSION 7700,13; FOR HELP, READ SYS:FAILSA,
HLP, OR
TYPE /H
*/W
**.*
5-SERIES FAILSAFE TAPE RECORDED BY VERSION
77 AT 10:47:50  03 AUG 72
"TAPE # 01

SEARCHING FOR 20,1547

FOUND ON TAPE    DSKC 20,1547

*
```

3. Listing a user's directory on the terminal

```
.MOUNT MTA; FAILSA /VID;FAILSAFE-TAPE  
OPERATOR NOTIFIED  
WAITING,..  
FAILSA (MTA0) MOUNTED
```

```
.R FAILSA
```

```
REWINDS ARE NOT AUTOMATIC  
FAILSAFE VERSION 7700,13; FOR HELP, READ SYS;FAILSA,  
HLP, OR  
TYPE /H  
*/W  
*/L
```

```
5-SERIES FAILSAFE TAPE RECORDED BY VERSION  
77 AT 10:47:50 03 AUG 72  
"TAPE # 01
```

```
SEARCHING FOR 20,A547
```

```
DIRECTORY FOR DSKC           20,1547
```

```
F40    .CTL  
DDT34 .MAC  
CTL    .TEC
```

```
BASIC .CTL  
COMMAND.CTL  
FAILSA,CT1  
FAILSA,OLG  
FAILSA,CTL  
FAILSA,LOG  
FILE1 .MAC  
FILE2 .CBL  
FILE3 .MAC  
PPNTST,MAC  
PPNTST,REL
```

```
*
```

## FILCOM Program

### Function

The FILCOM program is used to compare two versions of a file and to output any differences. Generally, this comparison is line by line for ASCII files or word by word for binary files. FILCOM determines the type of comparison to use by examining either the switches specified in the command string or the extensions of the files. Switches always take precedence over file extensions.

### Command Format

.R FILCOM

\*output dev:file.ext [directory] = input dev(1):file.ext [directory],  
input dev(2):file.ext [directory]

output dev: = the device on which the differences are to be output.

input dev: = the device on which an input file resides.

### Defaults

1. If the entire output specification is omitted, the output device is assumed to be TTY. However, the equal sign must be given to separate the input and output specifications of the command string.
2. If an output filename is specified, the default output device is DSK.
3. If the output filename is omitted, the second input filename is used, unless it is null. In this case, the filename FILCOM is used.
4. If the output extension is omitted, .SCM is used on a source compare and .BCM is used on a binary compare.
5. If the [directory] is omitted (input or output side), the user's default directory is assumed.
6. If an input device is omitted, it is assumed to be DSK.
7. If the filename and/or extension of the second input file is omitted, it is taken from the first input file.
8. A dot following the filename of the second input is necessary to explicitly indicate a null extension, if the extension of the first input file is not null. For example, to compare FILE.MAC and FILE. (i.e., with null extension), use the following command string:

```
.R FILCOM  
* = FILE.MAC.FILE.
```

9. The second input file specification cannot be null unless a binary compare is being performed. In a binary compare, if the first input file is not followed by a comma and a second input file descriptor, the input file is compared to a zero file and is output in its entirety. This gives the user a method of listing a binary file. Refer to Example 4.

#### Switches

The following switches can appear in the command string, depending on whether a source compare or a binary source compare is being performed.

##### Binary Compare

- |     |  |
|-----|--|
| /H  | Type list of switches available (help text from device SYS:).  |
| /nL | Specify the lower limit for a partial binary compare (n is an octal number). This switch, when used with the /nU switch, allows a binary file to be compared only within the specified limits.   |
| /Q  | When the files are different, print the message ?FILES ARE DIFFERENT, but do not list the differences. This switch is useful when BATCH control files want to test for differences but do not want the log file filled with these differences. |
| /nU | Specify the upper limit for a partial binary compare (n is an octal number). This switch, when used with the /nL switch, allows a binary file to be compared only within the specified limits.   |
| /W  | Compare files in binary mode without expanding the files first (refer to Appendix D). This switch is used to compare two binary files with ASCII extensions.   |
| /X  | Expand SAV files before comparing them in binary mode. This action removes differences resulting from zero compression (refer to Appendix D).  |

##### Source Compare

- |    |   |
|----|---|
| /A | Compare files in ASCII mode. This switch is used to force a source compare on two ASCII files.  |
| /B | Compare blank lines. Without this switch, blank lines are ignored.  |
| /C | Ignore comments (all text on a line following a semicolon) and spacing (spaces and tabs). This switch does not cause a line consisting entirely of a comment to become a blank line, which is normally ignored. |

/H	Type list of switches available (help text from device SYS:).
/nL	Specify the number of lines that determine a match (n is an octal number). A match means that n successive lines in each input file have been found identical. When a match is found, all differences occurring before the match and after the previous match are output. In addition, the first line of the current match is output after the differences to aid in locating the place within each file at which the differences occurred. The default value for n is 3.
/Q	Print the message ?FILES ARE DIFFERENT, when the files are different, but do not list the differences.
/S	Ignore spaces and tabs.
/U	Compare in update mode. This means that the output file consists of the second input file with vertical bars (or back slashes for 64-character printers) next to the lines that differ from the first input file. This feature is useful when updating a document because the changes made to the latest edition are flagged with change bars in the left margin. The latest edition of the document is the second input file.

If switches are not specified in the command string, the files are compared in the mode implied by the extension. The following extensions are recognized as binary and cause a binary compare if one or both of the input files have one of the extensions.

.BAC	.HGH	.RMT
.BIN	.LOW	.RTB
.BUG	.MSB	.SAV
.CAL	.OVR	.SFD
.CHN	.QUE	.SHR
.DAE	.QUF	.SVE
.DCR	.REL	.SYS
.DMP	.RIM	.UFD
		.XPN

Binary files are compared word by word starting at word 0 except for the following two cases:

1. Files with extensions .SHR and .HGH are assumed to be high segment files. Since the word count starts at 400000, upper and lower limits, if used, must be greater than (or equal to in the case of the lower limit) 400000.
2. Files with extensions .SAV, .LOW, and .SVE are assumed to be compressed core image files and are expanded before comparing.

Conflicts are resolved by switches or defaults. If a conflict arises in the absence of switches, the files are assumed to be ordinary binary files.

## Output

In most cases, headers consisting of the device, filename, extension, and creation date of each input file are listed before the differences are output. However, headers do not appear on output from the /U switch (update mode on source compare).

Source compare output - After the headers are listed, the following notation appears in the left column of the output

n)m

where

n is the number of the input file, and  
m is the page number of the input file (see examples).

The right column lists the differences occurring between matches in the input files. Following the list of differences, a line identical to each file is output for reference purposes.

The output from the /U switch differs from the above-described output in that the output file created is the second input file with vertical bars in the left column next to the lines that are different from the first input file.

Binary compare output - When a difference is encountered between the two input files, a line in the following format appears on the output device:

octal loc. first file-word second file-word XOR of both words

If the exclusive OR (XOR) of the two words differs only in the right half, the third word output is the absolute value of the difference of the two right halves. This usually indicates an address that changed.

If one input file is shorter than the other, after the end of file is encountered on the shorter file, the remainder of the longer file is output.

## Characteristics

The R FILCOM command:

Places the terminal in user mode.

Runs the FILCOM program, thereby destroying the user's core image.

## Associated Messages

Refer to Chapter 4.

Examples

1. The user has the following two ASCII files on disk:

First File

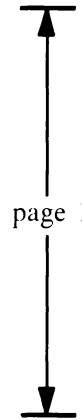
**FILE A**

A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M

Second File

**FILE B**

A  
B  
C  
G  
H  
I  
J  
1  
2  
3

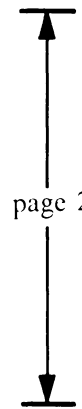


First File

N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z

Second File

N  
O  
P  
Q  
R  
S  
T  
U  
V  
4  
5  
W  
X  
Y  
Z



To compare the two files and output the differences on the terminal, the following sequence is used:

```
_R FILCOM )
```

Run the FILCOM program.

```
* = FILEA,FILEB )
```

Compare the two files on disk and output the differences on the terminal. By default, three consecutive identical lines determine a match.



```

headers → { FILE 1) DSK:FILEA      CREATED: 1456 17-JAN-1972
             FILE 2) DSK:FILEB      CREATED: 1456 17-JAN-1972

```

```

line
identical
in both
files → 1)1      FILE A
        1)      A
        ****
        2)1      FILE B
        2)      A
        *****
        1)1      D
        1)      E
        1)      F
        1)      G
line
identical
in both
files → 1)      G
        ****
        2)1      K
        *****
        1)1      L
        1)      M
        1)      N
line
identical
in both
files → 1)2      1
        ****      2
        2)1      3
        2)      N
        2)2
line
identical
in both
files → 1)2      W
        ****
        2)2      4
        2)      5
        2)      W
        *****

```

↑ This column indicates the page number of the file.  
 ↑ This column indicates either the first file or the second file.

To compare the two files and output the differences on the line printer, the following commands are used. Note that in this example the number of successive lines that determines a match has been set to 4 with the /4L switch.

```

_R FILCOM)
#LPT1:/4L = FILEA,FILEB)

```

```

FILE 1) DSK:FILEA      CREATED: 1456 17-JAN-1972
FILE 2) DSK:FILEB      CREATED: 1456 17-JAN-1972

```

```

1)1      FILE A
1)       A
1)       B
1)       C
1)       D
1)       E
1)       F
1)       G
****
2)1      FILE B
2)       A
2)       B
2)       C
2)       G

```

These lines are listed as being different because the /4L switch specifies that 4 consecutive lines must be found identical in the two files before they are considered as a match.

```

*****
1)1      K
1)       L
1)       M
1)2      N
****
2)1      1
2)       2
2)       3
2)2      N
*****
1)2      W
****
2)2      4
2)       5
2)       W
*****

```

To compare the two files so that the second input file is output with vertical bars in the left column next to the lines that differ from the first input file, use the following command sequence.

```

_R FILCOM)
#LPT1:/U = FILEA,FILEB)

```

```

| FILE B
|
| A
| B
| C
| G
| H
| I
| J
| 1
| 2
| 3
| N
| O
| P

```

The lines with vertical bars indicate the differences between the two files.

Q  
R  
S  
T  
U  
V  
| 4  
| 5  
W  
X  
Y  
Z

The lines with vertical bars indicate the differences between the two files.

- To compare two binary files on the disk and output the differences on the terminal, use the following command sequence.

```
.R FILCOM)
#TTY:_DSK:DIAL,REL,DIAL2)
FILE 1) DSK:DIAL,REL   CREATED: 0000 23-DEC-1971
FILE 2) DSK:DIAL2,REL  CREATED: 0000 12-AUG-1971

000000 000004 000001 000004 000060          000057
000002 000000 054716 000311 372712 000311 326004
000003 000006 000001 017573 510354 017575 510355
000004 000000 000000 017573 513216 017573 513216
```

- To compare two high segment files, the command sequence below is used. Note that the locations begin at 400000.

```
.R FILCOM)
#TTY:_SYS:TABLE,SHR, TABLE,SHR)
FILE 1) SYS:TABLE,SHR  CREATED: 2020 24-JAN-1972
FILE 2) DSK:TABLE,SHR  CREATED: 1829 30-NOV-1971

400000 001611 400010 001630 407157 000021 007147
400003 006675 000000 015024 407670 013651 407670
400004 005600 000070 004700 000113 001100 000163
400005 545741 444562 554143 625700 011602 261262
400010 634000 000000 260740 403516 454740 403516
400011 474000 000000 200000 414036 674000 414036
400012 402000 000156 202000 000720 600000 000676
400013 200040 406354 201000 000472 001040 406726
```

- To list a binary file, use the following command sequence.

```
.R FILCOM)
#TTY:_SYS:DOT,REL)
000000 000004 000001
000001 000000 000000
000002 000000 054716
000003 000006 000001
000004 000000 000000
000005 000007 517716
000006 000001 000002
000007 000000 000000
.
.
.
```

Note that the following sequence will not work because of the terminating comma.

```
*TTY:_SYS:DOT,REL,)
```

```
?COMMAND ERROR
```

5. To compare two binary files between locations 150-160 (octal).

```
._R FILCOM)
*TTY:/150/160U:_SYS:SYSTAT.SAV,_SYS:SYSDPY.SAV)
FILE 1) SYS:SYSTAT.SAV   CREATED: 0818 30-NOV-1971
FILE 2) SYS:SYSDPY.SAV   CREATED: 1642 29-NOV-1971

000150 200400 000137 200740 003217 000340 003320
000151 260740 004226 404500 004242 664240 000064
000152 260740 004253 66 002000 401240 006253
000153 200040 005011 260740 002723 060700 007732
000154 260740 004063 200040 004243 060700 000220
000155 201041 777777 202040 003241 003001 774536
000156 047040 000042 200040 004241 247000 004203
000157 254000 000174 251040 004142 005040 004036
000160 476000 006774 211040 000144 667040 006630
```

6. To compare two .SAV files. Note that the files are expanded before the comparison.

```
._R FILCOM)
*TTY:_SYS:TRY1.SAV,_SYS:TRY.SAV)
FILE 1) SYS:TRY1.SAV     CREATED: 2043 05-JAN-1972
FILE 2) SYS:TRY.SAV      CREATED: 0818 30-NOV-1971

000114 004000 000140 000000 000000 004000 000140
000116 777536 005536 000000 000000 777536 005536
000117 000000 005536 000000 000000 000000 005536
000120 006000 000140 007222 000140 001222 000000
000121 000000 006000 000000 007222 000000 001222
000130 010000 000005 000000 000000 010000 000005
000133 003727 005777 006643 007777 005164 002000
000137 003400 000070 046700 000004 045300 000074
000140 264000 001454 047000 000000 223000 001454
000141 260040 001773 200040 005075 060000 004706
000142 201240 001447 402000 006644 603240 007203
000143 542240 001634 251040 007221 713200 006416
000144 260040 002774 403000 000015 663040 002761
000145 621000 000010 476000 006715 257000 006705
000146 200240 003504 200740 006606 000500 005302
000147 251240 000012 051140 005076 200300 005064
000150 402000 003613 200400 000137 602400 003724
000151 201040 003730 260740 004226 061700 007516
000152 200260 003632 260740 004253 060520 007461
000153 321240 000164 200400 005011 121200 005175
```

## Function

The FILE command provides remote control of DECTape-to-disk and disk-to-DECTape transfers on operator-handled DECTapes.

## Command Formats

### 1. FILE C

Checks the queue of FILE commands to be read to determine if any of the user's requests are still pending. No argument is required. Pending requests will be listed.

### 2. FILE D, id, file.ext, file.ext, ...

Deletes the specified files from DECTape. Requires Tape ID and list of filenames as arguments. The tape ID is any alphanumeric name of 6 characters or less that is used to identify the tape. Upon completion, an automatic FILE L is performed.

### 3. FILE F, id, file.ext, file.ext, ...

Files information onto a DECTape. Requires Tape ID and list of filenames as arguments. Upon completion, an automatic FILE L is performed.

### 4. FILE L, id

Reads the directory of a DECTape and places it in the user's disk area as an ASCII file with filename id.DIR. id is any alphanumeric name of 6 characters or less that is used to identify the tape. It is the only argument. The user may then read the directory with a monitor command string. (See Examples).

### 5. FILE R, id, file.ext, file.ext, ...

Recalls (transfers) information from the user's DECTape to the disk. Requires Tape id and list of filenames as arguments. If the specified files already exist, they are superseded with the ones from the DECTape. If the specified files do not exist, they will be created on the first file structure in the job's search list for which creation is allowed. After the files are transferred, an automatic FILE L is performed.

### 6. FILE W

Waits until all of the user's pending requests are processed before continuing. If there are pending requests, the message WAITING ... is

typed to the user. Control returns when all requests have been processed. The user may type control-C if he decides not to wait.

7. FILE Z, id, file.ext, file.ext, ...

Zeros the directory of the DECTape before the files are copied and then performs the same operations as the F option. Requires Tape id and may have a list of filenames as arguments. After the files are copied, an automatic FILE L is performed.

The C and W functions are the only requests that are performed whenever possible. The user's terminal and job are free to proceed before the request is completed. The function argument is optional. If the function argument is not specified, a brief dialogue is performed.

In most cases the user does not need to specify which file structures the files are on because UMOUNT determines this (with LOOKUPS) and passes the information to OMOUNT.

However, file structure names may be specified in file descriptors. When no structure name is explicitly typed, the default is initially the first file structure in the user's search list (implied by DSK:) on which he is allowed to create files. Refer to the description of the SETSRC program. When a file structure name is typed or implied, it becomes the new default.

The asterisk construction may be used, but care should be taken when generic DSK: is typed. Because DSK: may define many file structures, the single file structure is chosen as follows:

When the asterisk construction is used for the filename or extension, the first structure on which the user may create files in his search list is used. This is called the user's standard file structure.

If the asterisk construction is not used and the file exists, the first file structure in the search list that contains the specified file is used, unless overridden by a default. (See Examples.) If the file does not exist, the standard structure is used.

#### WARNING

If the user has a search list with multiple file structures, the asterisk construction when used with the FILE R command can cause files to be created rather than superseded.

#### Characteristics

The FILE command:

Leaves the terminal in monitor mode.

Runs the UMOUNT program, thereby destroys the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

## Restrictions

The project-programmer number may not be specified in file descriptors.

## Associated Messages

Refer to Chapter 4.

## Examples

**.FILE R,MINF,MAIN,F4,SUBFIL,MAC)**

The files MAIN.F4 and SUBFIL.MAC are taken from the user's DECTape labeled MINE and placed on the first file structure in the user's search list for which creation is allowed.

There are two commands in the queue (counting this one).

**REQUEST STORED**  
**2,COMMANDS IN QUEUE**

**.FILE C)**  
**2, R JOB24 TTY5 27,235 MINE DSKB:,DSKB:MAIN,F4,SURFIL,MAC**  
**3, COMMANDS IN QUEUE**

The user checks to see if this request is still waiting to be processed. The first line of the output indicates that the user's request is second in the queue (2.), that the request made is a RECALL (R), that the user's job is 24 (JOB24), that the user is on terminal 5 (TTY5) under the project-programmer number of [27.-235] (27,235), that the tape is identified by the name MINE (MINE), and that the files will be written in the directory on DSKB: (DSKB:). The second line indicates that there are 3 commands in the queue.

**.FILE L,)**

The user wants the directory on the DECTape labeled 4 to be placed in his disk area as an ASCII file.

**.TYPE 4,DIR)**

The user then reads the directory file with the TYPE command.

If the user's search list is as follows:

DSKA/N,DSKB,DSKC

with file A on DSKA, file B on DSKB, and file C on DSKC, the following commands are equivalent:





## Function

The FILEX program is a general file transfer program used to convert between various core image formats, and to read and write various DECtape formats. Files are transferred as 36-bit data. The only processing on the data is that necessary to convert between various core image representation.

## Command Format

.R FILEX

\*dev:ofile.ext [directory] <nnn>/switches = dev:

file.ext [directory] /switches

If the project-programmer and/or the switches appear after the device name, they apply to all the following files. If they appear after the filename, the specifiers apply only to the preceding file. The input filename or extension may be \* in which case the usual processing of the \* construction occurs (refer to the TYPE command). The output filename and extension may be \* in which case the filename and extension of the input file is copied. If the output filename or extension is missing, the same procedure occurs as with the \* construction, except that all core image files are written with the default extension and format appropriate to the output device (unless overridden by switches).

If a protection <nnn> is not specified, files are written with the system standard protection unless the files are being written on SYS. On SYS, files are written with protection <155>, except for files with extension .SYS. These files have the default protection of <157>

Meaning of Switches:

Help text

/H - to obtain an explanation of the command string and individual switches.

DECtape Format Specifiers

/F - PDP-15 DECtape format

/M - MIT project MAC PDP-6/10 DECtape format

/O - Old DEC PDP-6 DECtape format

/T - normal PDP-10 directory format

/V - PDP-11 DECtape format (Note that PDP-11 contiguous files are not supported by FILEX.)

## FILE FORMAT SPECIFIERS

- /A - ASCII processing; meaningful only for PDP-11 and PDP-15 tapes.
- /B - binary processing; overrides default extension. Files read from a PDP-11 format tape with this switch contain four 8-bit bytes in each 36-bit word (1st byte in bits 10-17, 2nd byte in bits 2-9, 3rd byte in bits 28-35, and 4th byte in bits 20-27). Files written on a PDP-11 format tape with this switch are assumed to have the same format.
- /C - compressed; save file format. This format is assumed for files with extensions .SAV, .LOW, .SVE. The default output extension is .SAV unless the input extension is .LOW or .SVE, in which case the extension remains unchanged.
- /D - dump format. This format is assumed for files with extension .DMP.
- /E - expanded core image files (used by FILDDT). This format is assumed for files with extension .XPN. The default output extension is .XPN.
- /I - image processing; meaningful only for PDP-11 and PDP-15 tapes.
- /S - simple block (SBLK) format, project MAC's equivalent of .SAV format. The default output extension is .BIN.

## DECTape Processing Specifiers

- /G - (go on), ignores read errors on input device. FILEX checks the always-bad-checksum bit in the 5-series monitor, so this switch is not needed for files with .RPABC on (e.g., CRASH.SAV).
- /L - (list), causes a directory on an input DECTape file to be typed on the terminal, or causes a directory listing of the output DECTape at the end (i.e., after the output).
- /P - (preserved), causes quick processing (/Q) and preserves the scratch file after processing for use by another command.
- /Q - (quick), causes an input or output DECTape to be processed quickly by creating a scratch file on the disk. This file is deleted after processing is completed.
- /R - (reuse), reuses a scratch file preserved by a /P in a previous command.
- /Z - (zero), causes the appropriate format of a zeroed directory to be written on a DECTape output file. (Zeroing a DECTape directory is equivalent to deleting all the files on the tape.) If

TAPEID appears in the output specifier, then TAPEID is written as the tape identifier in the directory. TAPEID is preceded by a up arrow (^) and may be 6 characters on a PDP-10 tape, 3 characters on a project MAC tape, and is not present on a PDP-6 tape.

#### Characteristics

The R FILEX command:

Runs the FILEX program, thereby destroying the user's core image.

#### Examples

```
.R FILEX  
*DSK:_DTA1:TEST,DMP/C
```

The dump format file is compressed and written as TEST.SAV.

```
.R FILEX  
*DSK:SER105,XPn[10,1]_DSK:CRASH,SAV[1,4]
```

Copy CRASH.SAV to an expanded format file for FILDDT to examine.

## **FINISH Command**

### Function

The FINISH command terminates any input or output currently in progress on the specified device and automatically performs the RELEASE UOO (which CLOSES the files) and DEASSIGN command, thus making the device available to another user. This command is preferred over the DEASSIGN command because it completely disassociates an INITed device from the user's job, thereby preventing the user from continuing his program. If the user wishes to continue his program, he should use the DEASSIGN command.

### Command Format

FINISH dev

dev = the logical or physical name of the device on which I/O is to be terminated. This argument is optional.

If dev is omitted, I/O is terminated on all devices, except the job's controlling terminal and the logical name of the controlling terminal is cleared.

### Characteristics

The FINISH command:

Leaves the terminal in monitor mode.

Requires core.

Depends on FTFINISH which is normally absent in the DECsystem-1040.

### Restrictions

The user cannot continue his program if the device was INITed, but he can start at the beginning or enter DDT.

### Associated Messages

Refer to Chapter 4.

Examples

1 FINISH CDRI )  
2 FINISH DTA7I )  
3 FINISH LPTI )  
4

## **FUDGE Command (1)**

### Function

The FUDGE command creates a library REL file from a temporary file generated by a previous COMPIL, LOAD, EXECUTE, or DEBUG command string containing the /FUDGE switch. The library is created with the REL files in the same order in which they were specified in the command string. (Refer to the FUDGE2 program writeup and the LOADER documentation for descriptions of library REL files.)

### NOTE

Since the COMPIL program sorts out files by compilers, mixed FORTRAN and MACRO programs are sorted so that all FORTRAN programs are compiled first and MACRO programs second. However, the /FUDGE switch combines them in the order in which the COMPIL program encountered them.

When the /FUDGE switch is used, the user must issue the FUDGE command before issuing any other COMPIL-class command that runs PIP (e.g., TYPE,COPY). The reason for this is that there is only one temporary file containing information generated by COMPIL-class commands that run PIP. Therefore, the information generated by the /FUDGE switch will be superseded by the information generated by any other COMPIL-class command which runs PIP. By issuing the FUDGE command, the library REL file can be generated before the contents of the temporary file are overwritten by another COMPIL-class command.

### Command Format

FUDGE

### Characteristics

The FUDGE command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

---

1

This command runs the COMPIL program, which interprets the command before running the PIP program.

## Associated Messages

Refer to Chapter 4.

## Examples

```
._COMPIL/FUDGE;LIBRARY/MACRO TEST,MATE,DATPRO,CBL,SCIENC,F4)
```

Create a disk file named LIBRARY which contains the names of all the REL files produced.

```
._FUDGE)
```

Create the library file and call it LIBRARY. This file contains the following: TEST.REL, MATH.REL, DATPRO.REL, and SCIENC.REL.

## FUDGE2 Program

### Function

The FUDGE2 program is used to update files containing one or more relocatable binary modules and to manipulate the individual modules within these files. Relocatable binary modules are output by MACRO-10, F40, FORTRAN-10, COBOL, ALGOL, and BLISS-10. A module can be a complete program or only a set of subroutines. One reason for collecting a group of relocatable modules into one file is to enable LINK-10 or LOADER to use the file as a library (refer to the LINK-10 or LOADER documentation). Three files are used in the updating process:

1. A master file containing the file to be updated.
2. A transaction file containing the modules to be used when updating.
3. An output file containing the updated file.

All three files can be on the same device if the device is DSK. The two input files can be on the same DECTape.

The desired function of FUDGE2 is specified by a command code at the end of the command string. Only one command code can be specified in each command string. Switches can also be used to position a magnetic tape and to zero a DECTape directory (zeroing a DECTape directory is equivalent to deleting all the files on the tape).

### WARNING

For execution to occur, the command string must be terminated with an ALTmode, represented in this manual by a dollar sign (\$), instead of the usual carriage return-line feed.

### Command Format

```
.R FUDGE2
*output dev:file.ext = master dev:file.ext<modules>,
transaction dev:file.ext<modules>
(command)$
```

output dev: = the device on which the updated file is written. If omitted, DSK is assumed.

master dev: = the device containing the file to be updated. If omitted, the default is DSK. A comma is used to separate the master file and the transaction file.



transaction dev: = the device containing the modules to be used in the updating process. When more than one file is transferred from magnetic tape or paper tape, a colon must follow the device name for each file. For example,

MTA: : : TRANSFER 3 FILES

If the device is omitted, DSK is assumed.

file.ext = the filename and extension of each file. Filenames must be specified for directory devices, but the extension can be omitted. If the extension is not given, it is assumed to be .REL unless the /L switch appears in the command string. In this case, the output extension .LST is assumed.

Project-programmer numbers appearing after a filename apply to that file only. If the project-programmer number appears before the filename, it applies to all subsequent files until another device is specified.

The protection code of the master file is given to the output file.

The asterisk convention can be used with the input files (refer to Paragraph 1.4.2.4).

<modules> = Names of modules (on DSK or DTA only) to be used in the updating process. They are grouped within angle brackets in the same order as they appear in the file and are separated by commas. When manipulating all the modules within a file, only the filename need be specified. Module names cannot appear for the output file.

(command) = Code for the function to be performed. This code can be either preceded by a slash or enclosed in parentheses and must appear at the end of the command string. Each command results in the updated file being output to the output device. The command codes are as follows:

A Append the specified modules in the transaction file(s) to the master file.

C Compress the master file by deleting local symbols. These symbols are included in relocatable binary modules primarily because of their usefulness in debugging

procedures. Large libraries of debugged routines, such as LIBOL, frequently have the local symbols deleted in order to save disk space and reduce the amount of I/O required during the loading process.

D Delete the specified modules from the master file.

E Extract the specified files and/or modules from the input files. The entire file is extracted if module names are not specified.

H Type the commands and switches available (help text from device SYS:).

I Insert modules from the specified transaction files into the master file. The modules from the transaction files are inserted immediately before the specified modules in the master file. A comma is used to separate the transaction files.

L List the names and lengths of all relocatable modules within a file. The length is in one of two forms:

low segment break, high segment break  
or program break, absolute break

The length of FORTRAN modules is not output.

The default filename for spooled output is the name of the master file.

R Replace the specified modules in the master file with the specified modules in the transaction file. The number of replacing modules must be the same as the number of modules to replace.

S List all the entry points within a module. These entry points are listed across the page. The default filename for spooled output is the name of the master file.

X Write index blocks into a library file on DECtape or disk. Indexes cannot be written on magnetic tape. Index blocks are used in a direct access library search (refer to the LOADER documentation). This command implies a C command.

The method of numbering the blocks within a file is different on DECTape and disk. This can cause problems with indexed library files that are created on one device and loaded from the other. The index in an indexed library contains the name of each module in the library along with the block number within the library file of the beginning of that module. On disk, the blocks of a file are numbered relative to the beginning of the file; thus, an index references the same blocks properly no matter where the file is placed on the disk. However, on a DECTape, the block numbers are established relative to the beginning of the tape. Therefore, the area of the tape on which the file resides determines the block numbers that will be used in the index. When the LOADER references an indexed library on a device different from the one on which it was created, the block numbers in the index may no longer point to correct location within the library. This problem can also arise when loading an indexed file that was created at one location on a DECTape and then was transferred to a different location on that tape or to another tape. To transfer indexed files to other devices and then to load them from that device, the index blocks should be deleted before transferring and recreated on the new device. Any FUDGE2 command which generates a new binary file deletes the index blocks and causes a warning message to be output. The /X switch writes the index blocks.

#### NOTE

An indexed library created on the disk will work properly no matter how many times it is transferred to and from other devices, such as DECTape and magnetic tape, as long as the library is restored to the disk for use by the LOADER or LINK-10.

Comments are included on the FUDGE2 command string by preceding the comment with a semicolon. All characters after the semicolon, except for the ALTmode, are ignored until the next line feed, vertical tab, or form feed character is read.

File directories can be manipulated and magnetic tapes positioned by including switches in the command string. These switches can appear anywhere in the command string and are preceded by a slash or enclosed in parentheses. The following switches are available:

- /B Backspace a magnetic tape on file.
- /K Advance a magnetic tape one file.
- /T Skip to the logical end of tape on a magnetic tape.
- /W Rewind a magnetic tape.
- /Z Clear the directory of the output DECTape.

## Characteristics

The R FUDGE2 command:

Places the terminal in user mode.

Runs the FUDGE2 program, thereby destroying the user's core image.

## Associated Messages

Refer to Chapter 4.

## Examples

```
._R FUDGE2 )  
*_LPT:=DTA1:LIB40(L)$
```

List all relocatable modules from the file LIB40.REL, located on DTA1, on the line printer.

```
*DSK:LIB4BB=DTA2:LIB4AA <EXP,3,EXP,3C>,  
DTA1:F4<EXP,3A,EXP,3B>(R)$
```

Replace modules EXP.3 and EXP.3C located in file LIB4AA.REL on DTA2, with modules EXP.3A and EXP.3B in File F4.REL on DTA1: write out the new LIB4AA file on disk and call it LIB4BB.REL.

```
*DTA1:NFILE=DSK:MFILE<M1,M2,M3,M4>,  
DTA3:TFILEA<TA1,TA2>,  
DTA4:TFILEB<TB1,TB2>/I$
```

Insert into MFILE the modules TA1 and TA2 from TFILEA, and TB1 and TB2 from TFILEB. Create NFILE with the following order:

TA1.M1.TA2.M2,  
TB1.M3.TB2.M4

Insertion is on a one-to-one basis. If there are more modules to be inserted than specified modules before which they are to be inserted, the extra files are ignored.

```
*DTA1:NFILE=DSK:MFILE<M1,M2,M3,M4>,  
DTA3:TFILEA,  
DTA4:TFILEB/I$
```

However, in this example (where TFILEA.REL and TFILEB.REL contain the modules TA1 and TA2 and TB1 and TB2 respectively) create an NFILE.REL with the following order:

TA1.TA2.M1.TB1,  
TB2.M2.M3.M4

**\*DTA2:TESTA=MTA1:(WK),MTA2: (ZA)\***

Clears the directory of DTA2; rewind MTA1 and advance the tape one file; append the first two program files from MTA2 to the second file on MTA1 and write out the resultant file on DTA2, calling it TESTA.REL.

**\*OUTPUT=LIBARY,DTA1:LIBARY<FILEY,FILEZ>/A\***

Append the modules FILEY and FILEZ contained in the file LIBARY.REL on DTA1 to the end of the file LIBARY.REL on disk. Write the new file on disk and call it OUTPUT.REL.

**\*NEWFIL=OLDFIL<TEST,SUBTRC,MULTI>,BASFIL<PROG,ROUTIN,ANSWER>,SUBFIL<MATH>(E)\***

Extract the specified modules from the files OLDFIL, BASFIL, and SUBFIL and create a new output file called NEWFIL. The order of the modules in NEWFIL is as follows: TEST, SUBTRC, MULTI, PROG, ROUTIN, ANSWER, MATH.

**\*NEWF40=DTA2:OLDF40<SUBTLE,DATFIL,ROUTINE>/D\***

Delete the modules SUBTLE, DATFIL, and ROUTINE from the file OLDF40.REL on DTA2 and create a new output file NEWF40.REL on disk containing the remainder of file OLDF40.

**\*NOIDX,REL=IDX,REL(A)\***

Delete index blocks from the file IDX.REL and write the remainder of the file on the output file NOIDX.REL. The Append command (A) generates a new binary file and therefore removes the index blocks.

**\*^C**

Return to the monitor.

## GET Command

### Function

The GET command loads a core image from a retrievable storage device but does not begin execution.

This command clears all of user core. However, programs should not count on this action and should explicitly clear those areas of core that are expected to contain zeroes (i.e., programs should be self-initializing). This action allows programs to be restarted by a ^C, START sequence without having to do another GET command.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (2 EOFs in a row).

### Command Format

```
GET dev-file.ext[proj,prog] core
```

The arguments and the defaults are the same as in the RUN command.

The extension applies to the low file, not the high file. An extension of .SHR, then .HGH, is assumed for the high file. If the user types an extension of .SHR or .HGH, the extension is treated as a null extension since .SHR and .HGH are confusing as low file extensions.

### Characteristics

The GET command:

- Leaves the terminal in monitor mode.

- Does not operate when the device is currently transmitting data.

### Associated Messages

Refer to Chapter 4.

Example

.GET SYS:PIP)  
JOB SETUP

.GET TEST)  
JOB SETUP

## GLOB Program

### Function

The GLOB program reads multiple binary program files and produces an alphabetical cross-referenced list of all the global symbols (symbols accessible to other programs) encountered. This program also searches files in library search mode, checking for globals, if the program file was loaded by the LOADER in library search mode (refer to the LOADER documentation).

The GLOB program has two phases of operation; the first phase is to scan the input files and build an internal symbol table, and the second, to produce output based on the symbol table. Because of these phases, the user can input commands to GLOB in one of two ways. The first way is to specify one command string containing both the output and input specifications. (This is the command string format most system programs accept.) The second is to separate the command string into a series of input commands and output commands.

### Command Formats

#### 1. R GLOB

outdev:file.ext[directory]=input dev:file.ext[directory].  
file.ext.....,dev:file.ext[directory]Ⓢ

#### 2. R GLOB

followed by one or more input commands in the form

dev:file.ext[directory],file.ext[directory].....  
dev:file.ext[directory]....

and then one or more output commands in the form

outdev:file.ext[directory]=Ⓢ

When the user separates his input to GLOB into input commands and output commands (Command Format #2), the input commands contain only input specifications and the output commands, only output specifications. Each output command causes a listing to be generated; any number of listings can be printed from the symbol table generated from the current input files as long as no input commands occur after the first output command. When an input command is encountered after output has been generated, the current symbol table is destroyed and a new one begun.

### Defaults

1. If the device is omitted, it is assumed to be DSK. However, if the entire output specification is omitted, the output device is TTY.



2. If the output filename is omitted, it is the name of the last input file on the line (Command Format #1) or is GLOB if the line contains only output commands (Command Format #2). The input filenames are required.
3. If the output extension is omitted, .GLB is used. If the input extension is omitted, it is assumed to be .REL unless the null extension is explicitly specified by a dot following the filename.
4. If the project-programmer number [proj,prog] is omitted, the user's default directory is used.
5. An ALTmode terminates the command input and signals GLOB to output the cross-referenced listing. In other words, a listing is not output until GLOB encounters an ALTmode. The ALTmode appears at the end of the command string shown in Command Format #1 or at the end of each output command shown in Command Format #2.

### Switches

Switches control the types of global listings to be output. Each switch can be preceded by a slash, or several switches can be enclosed in parentheses. Only the most recently specified switch (except for L, M, P, Q, and X, which are always in effect) is in effect at any given time. If no switches are specified, all global symbols are output. The following switches are available.

- /A Output all global symbols. This is the default if no switches are specified.
- /E List only erroneous (multiple defined or undefined) symbols.
- /F List nonrelocatable (fixed) symbols only.
- /H List the switches available (help text) from SYS:GLOB.HLP.
- /L Scan programs only if they contain globals previously defined and not yet satisfied (library search mode).
- /M Turn off library search mode scanning resulting from a /L switch.
- /N List only symbols which are never referenced.
- /P List all routines that define a symbol to have the same value. The routine that defines the symbol first is listed followed by a plus (+) sign. Subsequent routines that define the symbol are listed preceded by a plus sign.
- /Q Suppress the listing of subsequent definers that result from the /P switch.
- /R List only relocatable symbols.
- /S List symbols with non-conflicting values that are defined in more than one program.

/X Do not print listing header when output device is not the terminal, and include listing header when it is the terminal. Without this switch, the header is printed on all devices except the terminal. The listing header is in the following format:

FLAGS SYMBOL OCTAL VALUE DEFINED IN REFERENCED IN

Symbols listed are in alphabetical order according to their ASCII code values. The octal value is followed by a prime (') if the symbol is relocatable. The value is then relative to the beginning of the program in which the symbol is defined. Flags preceding the symbol are shown below.

- M Multiply defined symbol (all values are shown).
- N Never referred to (i.e., was not declared external in any of the binary programs).
- S Multiply specified symbol (i.e., defined in more than one program but with non-conflicting values). The name of the first program in which the symbol was encountered is followed by a plus sign.
- U Undefined symbol.

Characteristics

The R GLOB command:

Places the terminal in user mode.

Runs the GLOB program, thereby destroying the user's core image.

Associated Messages

Refer to Chapter 4.

Examples

**.R GLOB**

Run the GLOB program.

**\*LPT1=MAIN,DTA2:SUB40,SUB50**

All global symbols in the programs MAIN (on DSK), SUB40, and SUB50 (on DTA2) are listed on the line printer. Along with the symbol is listed its

value, the program in which it is defined, all programs in which it is referenced, and any error flags.

**\*DTA4: BATCH,REL,DTA,DTA6:NUMBER,REL,CLASS)**  
**\*DSK: MATH,REL,LIBARY.)**

The programs to be scanned are BATCH.REL, DATA.REL on DTA4; NUMBER.REL, CLASS.REL on DTA6; and MATH.REL, LIBARY.null on DSK.

**\*LPT: =/F**

List only nonrelocatable symbols on the line printer.

**\*DSK: SYMBOL =/R**

List only relocatable symbols in the file named SYMBOL in the user's default directory.

**\*TTY: =/E**  
**U EXTSYM SUBRTE**

Print all erroneous symbols on the terminal. EXTSYM is an undefined symbol appearing in the program SUBRTE.

**\*^C**

Return to monitor mode.

## GRIPLE Program

### Function

The GRIPLE program accepts text from a user and records it in a disk file, thereby enabling users to record comments and complaints to be read at a later time by the operations staff.

### Command Format

#### R GRIPLE

When the GRIPLE program responds with a YES?, type the text, using as many lines as necessary, terminated with an ESCAPE. The text is written as a file with <157> protection and includes a header with the date, time, and project-programmer number of the user writing the comment. Therefore, the user does not need to identify himself.

### Characteristics

The R GRIPLE command:

Places the terminal in user mode.

Runs the GRIPLE program, thereby destroying the user's core image.

### Associated Messages

Refer to Chapter 4.

### Example

```
␣R GRIPLE␣  
YES? (TYPE ESCAPE WHEN THROUGH) THIS CONSOLE IS  
ALMOST OUT OF PAPER&  
THANK YOU
```

␣

## HALT Command

### Function

The HALT (^C) command transmits a HALT command to the monitor command interpreter. It stops the job and stores the program counter in the job data area (.JBPC). Refer to the DECSYSTEM-10 Monitor Calls for a description of the job data area.

### Command Format

HALT (^C)

### Characteristics

The HALT (^C) command:

Places the terminal in monitor mode.

Does not require LOGIN.

### Example

^C

⋮

## HELP Command

### Function

The HELP command is used to obtain useful information on various system feature. The user can select this information from the following special disk devices: HLP:, NEW:, OLD:, and SYS:.

### Command Formats

1. HELP

Outputs the instructions for receiving information.

2. HELP dev:\*

Outputs both the names of features that have available documentation contained on the specified device and the names of monitor commands, including the SET commands.

3. HELP dev:name

Outputs the information on the named feature contained on the specified device.

Only the first six characters of the name argument are scanned. These characters must be A through Z, 0 through 9, or asterisk (\*).

If the requested file is not found on the specified device, or if the device argument is omitted, the devices are searched in the following order: HLP:, SYS:, NEW:, and OLD:. However, if the user has device NEW: enabled in this search list (refer to the SETSRC program writeup) when using Command Format #3, it is searched before device SYS:. Both logical and physical names are searched for each device.

### Characteristics

The HELP command:

Leaves the terminal in monitor mode.

Does not require LOGIN.

Associated Messages

Refer to Chapter 4.

Examples

.HELP \*)

HELP IS AVAILABLE FOR THE FOLLOWING:

ALLOT	BASIC	BATCON	BLIS1Ø	BOOT11	BOOTGT	CDPSPL	CDRSTK
CHANGE	COBDDT	COBOL	COBRG	CREDIR	CREP	DDT	DELFIL
DIRECT	DSKRAT	FAILSA	FGEN	FILCOM	FORDDT	FORTRA	FUDGE2
GASP	GLOB	HELP	INITIA	ISAM	JQ	LIBRARY	LINED
LINK	LOGIN	LPTSPL	MACRO	MACY11	MAIL	MATCH	MATHLA
MTCOPY	OMOUNT	OPSER	PIP	PLTSPL	PTPSPL	QUEUE	QUOLST
RERUN	RUNINP	RUNOFF	SETSRC	SORT	SOUP	SPACE	SPRINT
SUBNEW	SUBSYS	SUSDPY	SYSERR	SYSTAT	TECO	UMOUNT	

THE MONITOR HAS THE FOLLOWING COMMANDS:

ASSIGN	ATTACH	BACKSP	CCONTI	CLOSE	COMPIL	CONTIN	COPY
CORE	CPUNCH	CREATE	CREP	CSTART	CTEST	D	DAYTIM
DCORE	DDT	DEASSI	DEBUG	DELETE	DETACH	DIRECT	DISMOU
DSK	DUMP	E	EDIT	EOF	EXECUT	FILE	FINISH
FUDGE	GET	HALT	HELP	INITIA	JCONTI	KJOB	LABEL
LIST	LOAD	LOCATE	LOGIN	MAKE	MOUNT	PJOB	PLEASE
PLOT	PRESER	PRINT	PROTEC	PUNCH	QUEUE	R	REASSI
REENTE	RENAME	RESOUR	REWIND	RUN	SAVE	SCHEDU	SEND
SET	SKIP	SOS	SSAVE	START	SUBMIT	SYSTAT	TECO
TIME	TPUNCH	TTY	TYPE	UNLOAD	USESTA	VERSIO	WHERE
ZERO							

THE MONITOR HAS THE FOLLOWING SET COMMANDS:

BLOCKS	BREAK	CDR	CORMAX	CORMIN	CPU	CTEST	DATE
DAYTIM	DENSIT	DSKFUL	DSKPRI	HPQ	NOMESS	OPR	SCHEDU
SPOOL	TIME	TTY	WATCH				

THE MONITOR HAS THE FOLLOWING SET TTY COMMANDS:

ALTMOD	BLANKS	CRLF	DEBREA	ECHO	ELEMEN	FILL	FORM
GAG	IGNORE	LC	NO	PAGE	RTCOMP	SLACE	SPEED
TABS	TAPE	TIDY	US	WIDTH			

THE MONITOR HAS THE FOLLOWING SET WATCH COMMANDS:

DAT	MTA	READS	RUN	VERSIO	WAIT	WRITES	
-----	-----	-------	-----	--------	------	--------	--

.HELP NEW:\*\*)

HELP IS AVAILABLE FOR THE FOLLOWING:

BASIC    BATCON    FORTRA

THE MONITOR HAS THE FOLLOWING COMMANDS:

ASSIGN	ATTACH	BACKSP	CCONTI	CLOSE	COMPIL	CONTIN	COPY
CORE	CPUNCH	CREATE	CREP	CSTART	CTEST	D	DAYTIM
DCORE	DDT	DEASSI	DEBUG	DELETE	DETACH	DIRECT	DISMOU
DSK	DUMP	E	EDIT	EOF	EXECUT	FILE	FINISH
FUDGE	GET	HALT	HELP	INITIA	JCONTI	KJOB	LABEL
LIST	LOAD	LOCATE	LOGIN	MAKE	MOUNT	PJOB	PLEASE
PLOT	PRESER	PRINT	PROTEC	PUCH	QUEUE	R	REASSI
REENTE	RENAME	RESOUR	REWIND	RUN	SAVE	SCHEDU	SEND
SET	SKIP	SSAVE	START	SUBMIT	SYSTAT	TECO	TIME
TPUNCH	TTY	TYPE	UNLOAD	VERSIO	WHERE	ZERO	

THE MONITOR HAS THE FOLLOWING SET COMMANDS:

BLOCKS	CDR	CORMAX	CORMIN	CPU	CTEST	DATE	DAYTIM
DENSIT	DSKFUL	DSKPRI	HPQ	NOMESS	OPR	SCHEDU	SPOOL
TIME	TTY	WATCH					

.HELP DIRECT)

TYPE OUT=INPUT,INPUT,...  
 /ACCESS:N = ACCESS ALL LISTED FILES UNDER N BLOCKS LONG  
 /ALLOCATED = GIVE ALLOCATED LENGTHS  
 /BEFORE:DATE;TIME = JUST FILES CREATED BEFORE THIS DATE-TIME  
 /BLOCKS = GIVE LENGTHS IN BLOCKS (DEFAULT)  
 /CHECKSUM = COMPUTE CHECKSUM OF EACH FILE  
 /DETAIL = EVERYTHING FROM EXTENDED LOOKUP  
 /F = FAST MODE  
 /H = THIS TEXT  
 /L = OUT TO LPT  
 /N = NORMAL MODE  
 /NOCHECKSUM = OMIT CHECKSUMS (DEFAULT)  
 /NODETAIL = DON'T USE DETAIL MODE (DEFAULT)  
 /NOSORT = OMIT SORT OPTIONS (DEFAULT)  
 /NOSUMMARY = DON'T USE /SUMMARY MODE (DEFAULT)  
 /NOTITLES = OMIT TITLES (DEFAULT IF TTY:)  
 /NOUNITS = DON'T USE /UNITS MODE (DEFAULT)  
 /S = SLOW MODE  
 /SINCE:ATE\*TIME = JUST FILES CREATED SINCE THIS DATE-TIME  
 /SORT = OUTPUT IN FORMAT AMENABLE TO SORTING  
 /SUMMARY = JUST PRINT SUMMARY LINE  
 /TITLES = INCLUDE TITLES (DEFAULT IF TTY:)  
 /UNITS = GIVE ACTUAL UNITS FOR STRUCTURES  
 /W:N = TRY TO FILL PAPER WIDTH OF N COLUMNS  
 /WORDS = GIVE LENGTHS IN WORDS  
 /WRITTEN = GIVE LENGTHS WRITTEN (DEFAULT)

\* IS WILD NAME, ETC.  
 ? IS WILD LETTER OF NAME, ETC.  
 "OUT=" MAY BE OMITTED  
 DEFAULT IS TTY:;DIR=DSK:\*. \*[MY DIRECTORY]

ON MAGNETIC TAPE,  
 /DENSITY:200 OR 556 OR 800 = SELECT THE DENSITY  
 /EOTS = STOP AT DOUBLE FILE MARK (DEFAULT)  
 /FILES:N = STOP AFTER DOING N FILES (TAPE MARKS)  
 /MARKS = NOTE EACH END OF FILE MARK  
 /NOEOTS = DON'T STOP AT DOUBLE FILE MARK  
 /NOMARKS = DON'T ANNOTATE FILE MARKS (DEFAULT)  
 /NOREWINDS = DON'T REWIND TAPE  
 /PARITY:EVEN = READ TAPE IN EVEN MODE  
 /PARITY:ODD = READ TAPE IN ODDMODE (DEFAULT)  
 /REWINDS = REWIND TAPE BEFORE AND AFTER (DEFAULT)

.HELP NEW:FORTRA)

FILENAME:	FORTRA.HLP
DATE:	2 FEB 73
FUNCTION:	SUMMARY OF FORTRAN-10 COMMAND STRING SWITCHES.
SWITCH	MEANING
ADVANCE:N	ADVANCE MTA PAST N EOF MARKS.



BACKSPACE: N      BACKUP MTA UNTIL NTH EOF MARK,  
CROSSREF          GENERATE CROSS REFERENCE LISTING.  
DEBUG             COMPIL LINKS TO FORTRAN DEBUGGER.  
EXPAND            ADD OBJECT FILE OUTPUT TO LISTING  
INCLUDE           COMPILE AS IF D IN COLUMN 1 WERE SPACE,  
MACROCODE        ADD MACHINE LANGUAGE TRANSLATION TO LISTING.  
NOERRORS         DON'T TYPE ERROR MESSAGES ON TTY.  
OPTIMIZE         PERFORM GLOBAL OPTIMIZATION,  
TAPEND            ADVANCE MTA TO LOGICAL EOT,  
WIND              REWIND MTA OR DTA TO LOAD POINT.  
ZERO              CLEAR DTA DIRECTORY,  
[END OF FORTRA,HLP]

## INITIA Command

### Function

The INITIA command performs standard system initialization for the terminal issuing the command. This command is issued automatically at system startup and at the 400 series restart at certain designated terminals, but may be re-issued at any time by the user. This command is used to initiate specific system programs, such as the operator service program, OPSER, on a particular console.

When INITIA runs, it finds the file with the name TTY.INI, which is an ASCII file created by the user. TTY.INI contains user-specified options indicating the characteristics of the terminal(s) to be initialized. If TTY.INI cannot be found, INITIA cannot initialize any devices and a start-up message is printed.

### Command Format

INITIA

### Characteristics

The INITIA command:

Leaves the terminal in monitor mode.

Runs a specific system program

Does not require LOGIN.

Depends on FTCCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Examples

```
.INITIA )  
RX7A5 A SYS #514/546 16:22:44 TTY31 SYSTEM 514/546  
JOB 44 USER CUSTER,L.  
DSKB: PRODUCTION DSKB  
DSKC: PRODUCTION DSKC
```

.

## JCONTINUE Command

### Function

The JCONTINUE command forces a continue of the specified job if the job was in a ^C state because of a call to the device error message routine (HNGSTP).

### Command Format

JCONTINUE $n$

$n$  = the number of the job to be continued. This argument is required.

### Characteristics

The JCONTINUE command:

Places the terminal in monitor mode.

Does not require LOGIN.

Depends on FTJCON which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Example

```
_JCONT 14)
```

## KJOB Command

### Function

In multiprogramming systems, the KJOB command:

Stops all assigned I/O devices and returns them to the monitor pool.

Returns all allocated core to the monitor pool.

Returns the job number to the pool.

Leaves the console in the monitor mode.

Performs an automatic TIME command.

In swapping systems, the KJOB command performs all the above procedures. In addition, the command responds with

CONFIRM:

The user may type ^C to abort logout, or type an optional file structure name (or list of file structure names) preceded by one of the following:

- F) to logout immediately saving all files (including temporary files) as they are. Identical to R LOGOUT, or RUN UO to LOGOUT.
- D) to delete all files on the specified file structures. Responds with ARE YOU SURE? Type Y or D for YES, any other character for NO.
- K) to delete all unprotected files (i.e., files with Oxx protection code) on the specified file structures. If project 1 or other jobs are logged-in with the same project-programmer number, responds with ARE YOU SURE? Type Y or K for YES, any character for NO.
- P) to save and protect (i.e., assign a protection code of 1 in the owner's field) all but temporary files (TMP, CRF, LST) on the specified file structures. If project 1 or other jobs are logged-in with the same project-programmer number, responds with ARE YOU SURE? Type Y or P for YES, any other character for NO.
- S) to save without protecting all but temporary files on the specified file structures. If project 1 or other jobs are logged-in with the same project-programmer number, responds with ARE YOU SURE? Type Y or S for YES, any other character for NO.
- L) to list the directories of the specified file structures.
- I) to individually determine what to do with all files on the specified file structure as follows:

After each filename is listed type

- P) to protect the file.
- S) to save the file.
- K) to delete the file.
- Q) to learn if over logged-out quota on this file structure. If not over quota, nothing is typed, and the same filename is repeated.
- E) to skip to next file structure and save this file if below logged-out quota for this file structure. If not below logged-out quota, a message is typed and the same filename is repeated.
- H) to list responses and meanings.
- U) to individually determine what to do with all but protected files. Protected files are always preserved.
- B) to delete no files except when user is over the logged-out quota, then delete enough files to be below quota. The files are deleted in the following order: 1) unprotected files according to the category of the file, 2) spooled files not previously queued, and 3) protected files according to the category of the file. The categories of files are as follows: 1) temporary files, 2) relocatable files, 3) backup files, 4) save files, and 5) all other files.
- Q) to learn if over logged-out quota on the specified file structures.
- H) to list the KJOB options and their meanings.
- W) to list the names of the files that are deleted.
- X) to turn off the listing of names of the files that are deleted. Complement of W.

If no file structure names are specified, the responses are for all file structure names in the job search list. If file structure names are specified, the responses apply to those file structures, and CONFIRM is retyped. The KJOB command ignores all logical assignments.

The user has the option of going through the CONFIRM dialogue, even if other jobs are logged-in under the same project-programmer number or if he is logged-in under project 1. (However, if sufficient responses are included on the KJOB command line or in a temporary file entered through an alternate entry point, CONFIRM is not typed.) By responding to a CONFIRM message, the user has an opportunity to organize this disk area by deleting or preserving specific files.

The KJOB program calls the QUEUE program to perform the queuing of files which have been deferred to logout time. This includes all spooled output unless the user has specifically queued output spooling earlier. Queuing may be suppressed with the /Z response (see below).

## Command Formats

### 1. KJOB

#### CONFIRM:

When the CONFIRM: response is given, the user may type any of the above-described letters followed by an optional file structure name or list of file structure names separated by commas. The user may type one of the above-described letters, followed by optional file structure names, on the same line as the KJOB command, and the CONFIRM: message will not be typed.

2. KJOB<log file descriptor> = / <letter><list of file structure names><letter><list of file structure names>etc.

<log file descriptor> has the following form: <dev:file.ext[proj,prog]> If the log file is not a disk or spooled device, TTY is used.

<letter> = any letter from the above-described set. In addition, the following responses are available to any jobs using this command format:

/Z:n specifies the degree of queuing desired:

- n = 0 suppresses all normal queuing done at LOGOUT time.
- n = 1 queues the log file only.
- n = 2 queues the log file and spooled output (\*.LPT, etc.)
- n = 3 queues the log file, spooled output, and \*.LST.
- n = 4 queues the log file, spooled output, \*.LST, and any requests deferred to LOGOUT time (deferred requests are not yet implemented).

If Z is given without a value or if there are no spool bits set for job, Z:0 is assumed. Otherwise, /Z:2 is assumed.

/VL:n specifies that the limit of pages for LPT files is to be n (decimal).

/VC:n specifies that the limit of cards for CDP files is to be n (decimal).

/VT:n specifies that the limit of feet of paper tape for PTP files is to be n (decimal).

/VP:n specifies that the limit of minutes for PLT files is to be n (decimal).

/VR:n specifies that the priority of the queue request is to be n; n is from 0 through 62. /VR:62 is the standard.

/VS:n specifies that the sequence number for the queue request is to be n.

/VD:v specifies that the file disposition of the log file is to be v.

- v = D deletes the log file after printing.
- v = P preserves the log file after printing.
- v = R renames the log file before printing to the queue area and deletes it after printing.

Default is /VD:R.

If a value to the above switches is not specified, the value is equivalent to 0 (e.g., /VD is equivalent to /VD:0). For the /Vx switches, a value of 0 is equivalent to the standard (e.g., /VD=/VD:0=/VD:R).

The letters must appear on the input side of the command string. If the log file is specified, all TTY output is appended to the log file. If no log file is specified or if the log file is not a disk or spooled device, the default is TTY. In addition, if responses to CONFIRM are required and are not specified on the KJOB command line, these responses will then be read from TTY. Therefore, users should be careful when employing this command format.

3. The KJOB program may be entered at the CCL entry point through the RUN UJO. When this is done, TPCOR file KJO or disk file nnnKJO.TMP, where nnn is the user's job number in decimal, is used instead of the TTY input. This temporary file has the following format:

KJOB<log file descriptor> = / <list of file structure names>/  
<letter><list of file structure names> etc.

## Characteristics

The KJOB command:

Detaches the terminal.

Stops all assigned I/O devices since it does not operate when a device is currently transmitting data.

Runs the KJOB and LOGOUT programs

Does not require LOGIN.

## Associated Messages

Refer to Chapter 4.

Examples

1. An example of the CONFIRM dialogue.

```
.K)  
CONFIRM: I)  
DSKB:  
TEST4 ,TST <055> 2000, BLKS : K)  
TEST5 ,TST <055> 505, BLKS : P)  
T11 ,BAK <055> 5, BLKS : K)  
T2 ,BAK <055> 5, BLKS : K)  
T3 ,BAK <055> 5, BLKS : K)  
TEST ,BAK <055> 5, BLKS : K)  
TEST ,REL <055> 5, BLKS : S)  
TEST ,MAC <055> 5, BLKS : P)  
TEST ,SHR <055> 30, BLKS : S)  
JOB 5, USER [10,63] LOGGED OFF TTY24 AT 2309 11-MAY-71  
DELETED 5 FILES  
SAVED 4 FILES 2565 TOTAL BLOCKS USED  
RUNTIME 0 MIN, 00,60 SEC
```

2. An example of the user bypassing the CONFIRM dialogue.

```
.K/F)  
JOB 9, USER [10,110] LOGGED OFF TTY3 1349 18-MAR-71  
SAVED ALL 23 FILES (630, DISK BLOCKS)  
RUNTIME 1 MIN, 51,52 SEC
```

3. An example of the command when used in the Batch system. The output appears in the log file.

```
12:20:51 MONTR K DSKB0:MUM,LOG[10,110]=/W/B/VL:200  
12:21:02 LGOUT JOB 12, USER [10,110] LOGGED OFF TTY50 1221 18-MAR-71  
12:21:02 LGOUT SAVED ALL 38 FILES (1275, DISK BLOCKS)  
12:21:02 LGOUT ANOTHER JOB STILL LOGGED IN UNDER [10,110]  
12:21:02 LGOUT RUNTIME 0 MIN, 00,65 SEC
```

4. An example of the user specifying two switches.

```
.K/W/B)  
DELETED:  
MYFILE  
JOB 14, USER [20,275] LOGGED OFF TTY35 1454 13-APR-72  
DELETED 1 FILES (1022, DISK BLOCKS)  
SAVED 52 FILES (1735, DISK BLOCKS)  
RUNTIME 0 MIN, 02,60 SEC
```



## LABEL Command (1)

### Function

The LABEL command allows the user to write an identifier onto a DECTape. This command is implemented for the user's convenience. The identifier is stored on the tape itself and is displayed when the user lists a directory of the tape with the DIRECT command. The PIP and FILEX programs can also read and write DECTape labels. Each user should assign a unique label to each of his DECTapes in order to avoid confusing one tape with another.

### Command Format

LABEL dev: ^ name ^

dev: = a physical or logical name which represents a DECTape.

^ = the delimiter of the DECTape identifier, which may be any non-alphanumeric character.

name = a one-to-six character SIXBIT name to be used as the identifier. Any characters can be used except the delimiter.

If the identifier consists entirely of alphanumeric characters, the delimiters can be omitted.

The DECTape must be assigned to the user before this command is executed.

### Characteristics

The LABEL command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

---

1

This command runs the COMPIL program, which interprets the command before running PIP.

Examples

```
.LABEL DTA7:COBOL1 )  
.LABEL DTA3:"PIP,33" )  
.ASSIGN DTA2:LOGNAM )  
.LABEL LOGNAM:"TAPEID" )
```

## Function

The LIST command directs PIP to list the contents of named source file(s) on the line printer (LPT). The output goes either to LPT immediately or to the disk to be spooled to LPT if it is being spooled for this job. Refer to the QUEUE and PRINT commands. If the LPT is being spooled, the QUEUE program should be used instead of the LIST command since it saves time and disk accesses.

## Command Format

### LIST list

list = a single file specification or a string of file specifications separated by commas. A file specification consists of a device name, a filename and extension, and a directory name. This argument is required. When a directory name precedes the filename, it becomes the default for all succeeding files.

Switches can be passed to PIP by enclosing them in parentheses in the LIST command string. When COMPIL interprets the command string, it passes the switches on to PIP.

## Characteristics

### The LIST command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

## Associated Messages

Refer to Chapter 4.

---

1

This command runs the COMPIL program, which interprets the command before running PIP.

Examples

```
.LIST TEST.* )  
.LIST *.MAC )  
.LIST DTA4:A,B,C )
```

## Function

The LOAD command translates the specified source files if necessary (function of COMPILE command), runs the linking loader, and loads the .REL files generated. The language translator used is determined by the source file extension or by switches in the command string (refer to the COMPILE command). If a REL file already exists with a more recent date than that of the source file compilation is not performed (unless requested via a switch).

This command generates a core image but does not begin execution. At this point, the user can start his program or save the core image for future execution.

Each time the COMPILE, LOAD, EXECUTE, or DEBUG command is executed, the command with its arguments is remembered in a temporary file on disk, or in core if the monitor has the TMPCOR feature. Therefore, issuing one of these commands without any arguments causes the arguments saved in the temporary file to be reused (refer to Paragraph 1.5).

The LOAD command accepts several command constructions: the @ construction (indirect commands), the + construction, the = construction, and the <> construction. Refer to Paragraph 1.5 for a complete description of each of these constructions.

## Command Format

### LOAD list

list = a single file specification, or a string of file specifications separated by commas. A file specification consists of a device name, a filename with or without an extension, and a directory name (refer to Paragraph 1.4.2.4).

The following switches can be used to modify the command string. These switches can be temporary or permanent switches unless stated otherwise (refer to Paragraph 1.5.5).

/ALGOL	Compile the file with ALGOL. Assumed for files with the extension of .ALG.
/BIN	Generate a binary file for each file compiled. The name for the binary file follows the standard conventions for determining the name of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .REL. This is the default action.

---

1

This command runs the COMPIL program, which interprets the command before running the appropriate language translator and linking loader.

/BLISS(1)	Compile the file with BLISS10. Assumed for files with the extension of .B10 and .BLI.
/COBOL	Compile the file with COBOL. Assumed for files with the extension of .CBL.
/COMPILE	Force a compilation of this file even if a binary file exists with a newer date and time than the source file. This switch is used to obtain an extra compilation (e.g., in order to obtain a listing of the compilation) since normally compilation is not performed if the binary file is newer than the source file.
/CREF	Produce a cross-reference listing file on the disk for each file compiled for later processing by the CREF program. The filename for the listing file follows the standard conventions for determining the name of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .CRF. The files can then be listed with the CREF command. However, with COBOL files, the cross-referenced listing is always appended to the listing file. No additional command need be given to obtain the listing.
/FOROTS	Load the file with FOROTS (the new FORTRAN object-time system).
/FORSE	Load the file with FORSE (the old FORTRAN object-time system).
/FORTRAN	Compile the file with a FORTRAN compiler. Assumed for files with the extension of .F4 and .FOR and all files with nonrecognizable translator extensions if FORTRAN is the standard translator. This switch is needed if the file has a nonrecognizable translator extension and FORTRAN is not the standard translator or is not the current default (e.g., LOAD/COBOL, TEXT1, TEXT2, TEST3/FOR).
/FUDGE	Create a disk file containing the names of the .REL files produced by the command string. When the FUDGE command is given, PIP reads this file in order to generate a library REL file. Arguments to this switch are:

/FUDGE:dev:file.ext[proj,prog]

dev: - the device on which to write the file. If the device is omitted, DSK: is assumed.

file.ext - the name of the library file. The filename is required. If the extension is omitted, it is assumed to be .REL.

---

BLISS10 will be recognized as a language translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

[proj, prog] - the directory in which to place the file. The user's directory is assumed if none is given.

This switch is permanent in the sense that it pertains to all REL files generated by the command string.

- /F10            Use the FORTRAN-10 compiler when compiling the associated FORTRAN file. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.
- /F40            Use the F40 compiler when compiling the associated FORTRAN file. This is the current default action. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.
- /KA10  
/KI10           Designate the machine on which the program will execute once it has been loaded. These switches are needed for FORTRAN-10 programs since the compiler generates different code for the KA10 and KI10 processors. The /KA10 switch is the current default.
- /LIBRARY       Load the files in library search mode. This mode causes a program file in a special library file to be loaded only if one or more of its declared entry symbols satisfies an undefined global request in the source file. The default libraries are always searched. Refer to the LOADER and LINK-10 documentation.
- /LINK           Cause the file to be loaded by the LINK-10 linking loader. If used, this switch should be placed before any file specifications since the COMPIL program may have to generate load-control switches.
- /LIST           Generate a disk listing file, for each file compiled. The filename for the listing file follows the standard conventions for determining the name of the output file (refer to Paragraphs 1.5.2 and 1.5.3). The extension is .LST. These files can be listed later with the LIST command. If the line printer is being spooled for this job, the listing files are written on device LPT and are automatically spooled at LOGOUT time. Unless this switch is specified, listing files are not generated.
- /LMAP           Produce a loader map during the loading process (same action as /MAP) containing the local symbols.
- /LOADER        Cause the file to be loaded by the LOADER. This is the current default action.
- /MACRO         Assemble the file with MACRO. Assumed for files with extensions of .MAC.

/MACX11(1)	Assemble the file with MACX11. Assumed for files with an extension of .P11.
/MANTIS(2)	Compile the file with the MANTIS debugging information. This switch affects FORTRAN programs only.
/MAP	Produce a loader map during the loading process. When this switch is encountered, a loader map is requested from the loader. After the library search of the default libraries, the map is written in the user's disk area with either the filename specified by the user (e.g., /MAP:file) or with the default filename MAP.MAP if loading is performed by LOADER or nnnLNK.MAP if performed by LINK-10. This switch is an exception to the permanent compile switch rule in that it causes only one map to be produced although it may appear as a permanent switch.
/NEW	Run the appropriate language translator from the experimental system library (device NEW:) area [1,5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:. Refer to the following NOTE.
/NOBIN	Do not generate binary files. Unless this switch is given, binary files are generated. This switch, when combined with the /CREF or /LIST switch, is useful when compiling programs solely for the purpose of generating listings.
/NOCOMPILE	Complement the /COMPILE switch by not forcing a compilation on a source file whose date is not as recent as the date on the binary file. Note that this switch is not the same as the /REL switch which turns off all compilation, even if the source file is newer than the REL file. /NOCOMPILE is the default action.
/NOLIST	Do not generate listing files. This is the default action.
/NOMANTIS(2)	Compile the file without the MANTIS debugging information. This switch affects FORTRAN programs only.
/NOSEARCH	Load all routines of the file whether the routines are referenced or not. Since this is the default action, this switch is used only to turn off library search mode (/LIBRARY). This is not equivalent to the /P LOADER switch because /P does not search any libraries where /NOSEARCH will scan the default libraries.

---

1

MACX11 (the PDP-11 assembler for the PDP-10) will be recognized as a language translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

2

MANTIS will be recognized as the debugging program only if the appropriate assembly switch is set. However, this switch setting is not supported.



/OLD	Run the appropriate language translator from the system library of old programs (device OLD:) which resides on the disk area [1,3]. If the translator does not exist on device OLD:, try to obtain it from device SYS:. Refer to the following NOTE.
/REL	Use the existing .REL files although a newer source file may be present.
/SEARCH	The action is identical to that of the /LIBRARY switch.
/SELF	Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for an individual who keeps a private copy of a translator in his own disk area. System programmers occasionally keep experimental versions of standard translators in their disk area in order to test new features. Refer to the following NOTE.
/SNOBOL(1)	Compile the file with SNOBOL. Assumed for files with an extension of .SNO.
/SYS	Run the appropriate language translator from the system library (device SYS:) area [1,4]. This is the default action.

#### NOTE

Once a language translator has been specified from a particular area (e.g., /SELF), it cannot be called from a different area within the same command string, i.e., the following is illegal:

```
.LOAD ITEM.CBL/SYS,ITEM02.CBL/SELF
```

However, the following is valid:

```
.COMPIL ITEM.CBL/SYS
.COMPIL ITEM02.CBL/SELF
.LOAD /REL ITEM, ITEM02
```

#### Characteristics

The LOAD command:

Leaves the terminal in monitor mode.

Runs the appropriate language translator and linking loader, thereby destroying the user's original core image.

---

<sup>1</sup> SNOBOL will be recognized as a language translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

## Associated Messages

Refer to Chapter 4.

## Examples

```
LOAD TEST  
MACRO: TEST  
LOADING  
  
LOADER 2K CORE  
  
EXIT  
  
.
```

## LOCATE Command

### Function

The LOCATE command logically establishes the user's job at a specified station. When the job is initiated, the user's logical station corresponds to his physical station. Therefore, this command is needed only if the user desires to change his logical station.

### Command Format

LOCATE nn

nn = the station number.

An argument of 0 denotes the central station. A null argument implies the station of the user's terminal, i.e., his physical station.

### Characteristics

The LOCATE command:

Leaves the terminal in monitor mode.

Is valid only in systems with a remote station.

### Restrictions

The LOCATE command must specify a station that is currently in contact with the central station.

### Associated Messages

Refer to Chapter 4.

### Examples

```
⋮ LOCATE 2 )  
⋮ LOC 0 )  
⋮
```

## **LOGIN Command**

### Function

The LOGIN command is used to gain access to the system. This command loads a Monitor Support program which accepts the user's LOGIN data. The user types in his project and programmer numbers followed by his password. To login successfully, the project and programmer numbers and the password typed in by the user must match the project and programmer numbers and password stored in the system accounting file (SYS:ACCT.SYS).

### Command Format

LOGIN proj,prog

proj,prog = the user's project-programmer number. The project and programmer numbers may be separated by either a comma or a slash. If a slash is used, the message of the day is not output to the user unless the date on the file containing the message (NOTICE.TXT) is later than the last time the user logged-in. If this is true, the message is typed only once, whereas, when the comma is used, the message is output every time the user logs in. This argument may be typed on the same line as the LOGIN command, or on the following line after LOGIN types out the number sign.

### Characteristics

The LOGIN command:

Returns the terminal to monitor mode or starts a program running if specified in

ACCT.SYS entry for proj,prog.

Runs the LOGIN program.

Does not require LOGIN.

### Associated Messages

Refer to Chapter 4.

Example

The following is the procedure used to gain access to the system.

LOGIN 27,235)  
JOB 21 550417A TTY23

LOGIN types the job number assigned to user (job number 21), followed by monitor name, version number, and console line number. If the user does not type his project-programmer number on the same line as the LOGIN command, LOGIN outputs a number sign indicating that the user should type in his project-programmer number.

PASSWORD!

System requests user to type his password. User types password followed by carriage return (refer to Paragraph 1.4.2.1). To maintain password security, the monitor does not echo the password. On terminals with local-copy (refer to DECsystem-10 Monitor Calls), a mask is typed to make the password unreadable.

1135 8-JUN-71 THUR  
TYPE SYS,SCHED FOR NEXT  
WEEKS SCHEDULE  
.

If user entries are correct, the system responds with time, date, day of the week, the message of the day (if any), and a period, indicating readiness to accept another command.

## MAKE Command

### Function

The MAKE command creates a new file on the disk with TECO (Text Editor and Corrector). If a file already exists with the same name, a warning message is given and the file will be superseded if the user continues in TECO. If the user types two control-Cs to leave TECO, the original file will not be destroyed. Refer to the TECO manual in Notebook 6 of the DECsystem-10 Software Notebooks.

### Command Format

MAKE dev:file.ext [directory]

dev: = the device or file structure name on which the file is to be created. If omitted, DSK: is assumed.

file.ext = any legal filename and filename extension. The filename is required; the filename extension is optional.

[directory] = the directory in which the file is to be created. If omitted, the user's default directory is assumed. Note that the default directory may be an SFD or some other UFD.

Switches can be passed to TECO by preceding the switch with a slash in the MAKE command string. When COMPII. interprets the command string, it passes the switch on to TECO.

### Characteristics

The MAKE command:

Places the terminal in user mode.

Runs the TECO program, thereby destroying the user's core image.

Depends on FTCCLS which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

---

1

This command runs the COMPII. program, which interprets the commands before running TECO.

Example

```
._ MAKE TEST3,MAC )  
*
```

## **MOUNT Command**

### Function

The MOUNT command allows the user to request assignment of a device via the operator. This command is similar to the ASSIGN command, but, whereas the ASSIGN command operates without operator communication, the MOUNT command requests operator interaction when necessary. For example, if a Batch user requests a DEctape drive and all drives are in use, then the operator can free one for the user, if he wishes. With this command the user can request devices from the restricted pool of devices, which he cannot do with the ASSIGN command.

The MOUNT command gives the operator greater control over assignment of devices on the system. When a user requests a device via this command, the operator has the option of either selecting a specific unit (e.g., DTA5) or cancelling the request completely (all units of this type are in use and the operator does not want to free one for this user). The operator may also mount the media for the requested unit if the media is sufficiently identified (e.g., a deck of cards in the card reader or an identified DEctape on a specific drive).

When the MOUNT command is used to gain access to a file structure, it allows the user to specify a particular drive, places the file structure name at the end of the job's search list, and waits for completion of operator action, if desired. Each file structure can have an administrative file, QUOTA.SYS, which contains a list of quotas for all users allowed access to the structure. When the file structure is mounted, a UFD is created for the user if he has an entry in QUOTA.SYS on the file structure.

The MOUNT command runs the UMOUNT program in the user's core area. UMOUNT scans the command string and completes as much of the command as possible without operator intervention. When operator intervention is required, UMOUNT queues a request to the OMOUNT program by writing a command file on the [3,3] disk area. OMOUNT examines these command files and interacts with the operator. When the command file is deleted, the operator action has been completed. UMOUNT waits for this completion of operator action unless the user types a control-C. When a control-C is typed, the user does not receive a message of confirmation, but can later use the /CHECK switch to see if his request is still pending (see Examples).

To insure validity of any tape error analysis, MOUNT/DISMOUNT should be used to acquire and release magtape units. This mechanism provides the basis for all media-related error reporting.

### Command Format

MOUNT dev: logical-dev /switches (drives)

dev: = one of the following: (1) a physical device name (e.g., DTA3,CDR,MTA), (2) a logical name previously associated with a



physical device by either a MOUNT or ASSIGN command, or (3) a file structure name (one that is already mounted or one whose name appears in STRLST.SYS). This argument is required.

logical-dev = any SIXBIT name that is not the same as dev:. In other words, it may not be a physical device name or logical name that is currently being used or has previously been used as dev:. This argument is required when the device being mounted is a non-file structure device and is ignored when the device is a file structure.

switches = The following switches are optional and only enough characters to make the switch unique are required.

/CHECK	Check and list pending requests.
/HELP	Type this list.
/MULTI	Multi-access, disk only, complement of /SINGLE, default condition.
/PAUSE	Notify the user before sending the message to the operator for a request. The user can then abort the command if desired.
/REELID	A reel identification to be used when mounting a Magtape. The argument is a string of up to 6 characters which is used to uniquely identify a reel of Magtape. This identification will be included in any system generated error or status reports about the drive while the reel is mounted.
/ROONLY	Read only, same as /WLOCK.
/SINGLE	Only this job can access files on the structure (single access), file protection is enforced for him, disk only.
/VID:name	A visual identification passed to the operator as a comment to assist him in identifying a particular unit to mount. The argument can be in one of two forms: <ol style="list-style-type: none"><li>1. any string of up to 25 characters containing only letters, digits, periods, and hyphens, or</li><li>2. any string of up to 25 characters enclosed in single quotes. However, break characters and single quotes are not allowed in the string. The naming and use of this switch is relevant only to the extent that the installation operator knows what it</li></ol>

means. The PLEASE command should be used for any complex procedures or long communications with the operator.

/WENABL Write enable for this job, complement of /WLOCK, default condition.

/WLOCK Write locked for this job. This job cannot write on this file structure and the monitor will not update BAT blocks or the access date. If /SINGLE is given, the operator may set hardware write lock to ensure that nothing is written.

(drives) = the physical drives on which the units are to be mounted. A drive argument may be used only when mounting file structures. The drives must be in the logical unit order within the file structure. Drive names are separated by commas. Leading and embedded drives that are not specified must be represented by null names (.,DPA3). Unspecified trailing drives may be omitted. Drive names are as follows:

Blank, null - unspecified. UMOUNT finds one of proper type.

Two letters - controller class (e.g., DP).

Three letters - specific controller (e.g., DPA). UMOUNT finds a drive on that controller.

Three letters and one or two digits - specific drive (e.g., DPA0, DPA1).

The user, by specifying a drive list, may force the packs to be mounted on specific drives or controllers. If no drive (or incomplete) specification is given, an available drive of the proper type is found.

## Characteristics

The MOUNT command:

Places the terminal in user mode.

Runs the UMOUNT program, thereby destroying the user's core image.

## Associated Messages

Refer to Chapter 4.

Examples

U MOUNT PRIV )  
PRIV MOUNTED

Asks the operator to mount the file structure PRIV.

U MOUNT PAY:(DPA,,DPB)/S )

Requests that the first unit of file structure PAY be mounted on Controller A, the second unit on any controller, the third unit on controller B, and any remaining units on any drives. The structure will be single access (i.e., available only to this job).

U MOUNT MINE )  
OPERATOR NOTIFIED  
WAITING...

Mount the file structure MINE. The request is queued to the operator. UMount is waiting for the request to be completed. The user does not wait for confirmation.

\*C

.  
.  
.

U MOUNT/CHECK )

The user wants to know if his request has been processed.

NONE PENDING  
U R SETSRC )  
\*T  
DSKA,DSKP,PRIV,PAY,MINE,FENCE  
\*C  
U MOUNT DTA INPUT )

The request has been processed. The user wants to know if the file structure is in his search list. The file structure has been added to his search list. The user wants the operator to select a DECTape drive and assign it with logical name INPUT.

OPERATOR NOTIFIED  
WAITING...

The request is queued to the operator. UMount is waiting for the request to be completed.

INPUT (DTA5) MOUNTED  
U MOUNT INPUT/VID:325 )

The operator has selected DTA5. The user asks the operator to mount the DECTape labeled 325. He may use either DTA5 or INPUT to refer to his device. For example, the Batch user would use INPUT since he would not know what DECTape drive he is assigned.

OPERATOR NOTIFIED  
WAITING...  
INPUT (DTA5) MOUNTED  
U MOUNT INPUT OUTPUT )

The request is queued. UMount is waiting for confirmation. The mount is successful. The user changes the logical name to OUTPUT. The logical name INPUT is no longer valid.

OUTPUT (DTA5) MOUNTED

The mount is successful.

# OPSER Program

## Function

The OPSER program facilitates multiple job control from a single terminal by allowing the user to run several jobs called subjobs from his terminal. The OPSER program acts as the supervisor of the various subjobs by allowing monitor level or user level commands to be passed to all of the subjobs or to selected subjobs. Output from the various subjobs may be retrieved by OPSER.

The subjobs of OPSER run on pseudo-TTYs (refer to DECsystem-10 Monitor Calls) and all initializations of the pseudo-TTYs are performed by OPSER. The user needs only to provide the subjob name, either an OPSER-provided subjob number or a user assigned name. System programs, as well as user-written programs, that require a dedicated terminal can be run as subjobs of OPSER. By running jobs on pseudo-TTYs, OPSER is able to maintain an I/O link between the running jobs and the user. In addition, the output from the various subjobs is concentrated on one terminal instead of many, as was the case when each program required its own terminal.

The OPSER program is primarily intended for use by the system operator in controlling the separate components of the multiprogram batch facility. However, all users can employ this program for their own purposes. Refer to the MPB Operator's Manual in the DECsystem-10 Software Notebooks for complete information on OPSER. Also, Edition 3 of OPRCOM (Operator-Privileged Commands) contains a description of OPSER commands that are available only to the operator.

## Command Format

### R OPSER

OPSER signifies its readiness to process commands by typing an asterisk if no subjobs are in use or subjobs are in a wait for an operator action. OPSER responds with an exclamation point when a subjob is running. Commands may be entered whenever OPSER is operating. Each command is preceded by a colon and must be typed to sufficient length to make it unique.

### OPSER Commands

:AUTO/hh:mm filespec	Process the specified file as an automatic startup file. The execution of the file is terminated by an end-of-file or by the operator typing on the console. This is the normal way that the standard subjobs are started by the operator. The time argument is optional; when it is given, the AUTO file is run at the specified time. Comments can be
-------------------------	---

included in an AUTO file by preceding the comment with a semicolon. For example,

```
:SLOGIN  
:LOGIN A NEW SUBJOB
```

:CLOSE	Close the disk log file without opening a new one.
:CONTINUE	Continue processing the AUTO file after it has been interrupted by a Control-C. This allows the operator to gain control of a subjob during auto file processing.
:CURRENT	Type the number of the current subjob (the last one typed into). Output from another subjob does not affect current subjob.
:DAYTIME	Obtain the current date and time.
:DEFINE xxx=n	Associate the symbol xxx as the mnemonic for subjob number n. The symbol B is reserved for the subjob running BATCON.
:DEVICE nam:log:n	Assign the device with the physical name nam and logical name log to subjob n. The logical name is optional but a null field must be typed if the name is omitted, e.g., :DEVICE CDR::3.
:ERROR n,m,p	Report only error messages (that is, ignore nonerror messages from subjob n). Message reporting is resumed with the :REVIVE command.
:EXIT	Exit to the monitor if no subjobs are in use; otherwise give a list of those that are running. This should be used instead of ^C, since EXIT does not return the job to monitor mode if there are any active subjobs.
:FREE	Type the first free subjob number.
:HELP	Type a text which briefly explains the commands.
:JCONT n	Continue the specified stopped job.
:KJOB, n,m,p	Kill the specified subjobs saving all files. Causes /Z:0 to be included to KJOB so spooled files are not queued.
:KILL n,m,p	Kill the specified subjobs. This is identical to :KJOB.

:KSYS hmmm	Operator-privileged command. Refer to OPRCOM.
:LOGIN proj.prog	Login a new subjob. If no project-programmer number is typed, assume OPSER's project-programmer number.
:MSGVLVL 0	Cause the response to the :WHAT command to include the JOBSTS bits.
:MSGVLVL 1	Cause the response to :WHAT command to eliminate the JOBSTS bits.
:QUEUE <line>	Initiate the first free subjob and send the typed-in line to the system queue manager.
:RESOURCES	Type the list of the available system resources.
:RESTRICT dev:	Operator-privileged command. Refer to OPRCOM.
:REVIVE n	Resume normal echoing of output from subjob n.
:SEND ·line·	Simulate the SEND monitor command.
:SET a	Simulate a SET monitor command. Valid SET monitor commands are SET CORMAX, SET CORMIN, SET DATE, SET DAYTIME, SET OPR TTY, SET SCHED, and SET TTY.
:SET BATMAX n	Operator-privileged command. Refer to OPRCOM.
:SET BATMIN n	Operator-privileged command. Refer to OPRCOM.
:SET LOGMAX n	Operator-privileged command. Refer to OPRCOM.
:SET RUN	Operator-privileged command. Refer to OPRCOM.
:SILENCE n	Ignore all output from subjob n.
:SLOGIN proj.prog	LOGIN one subjob but suppress its response. If proj, prog is omitted, OPSER uses its own.
:STOP n	Put the specified subjob in monitor mode. This is equivalent to inputting two control-C's in interactive mode.
:SYSTAT xx	Run SYSTAT with optional argument xx over the first free subjob.

:TLOG filespec	Create a disk log file with the specified name. If the file already exists, a message is typed to determine whether the existing file should be superseded. If not, the file is appended to the existing one. Default for filespec is OPSER.LOG.
:TSILENCE n	Ignore all output from subjob n (same as the :SILENCE command) but place entries into the log file.
:TTYTST	Test this terminal by typing all the ASCII characters between octal 40 and 174, inclusive.
:UNRESTRICT dev:	Operator-privileged command. Refer to OPRCOM.
:WHAT n,m,p	Type the status of the specified subjobs on the terminal. The status includes a SYSTAT with the time, the time of the last input and the last output, a linear listing of the JOBSTS bits, and the time of the next timed auto file.

When a subjob number or name is required in a command string, the subjob may be specified in one of four ways. It can be omitted, in which case the last subjob typed into is used. The mnemonic ALL may be used, in which case all active subjobs are implied. A decimal number can be used from zero to the limit OPSER is generated for. Finally, a mnemonic can be assigned to the subjob with the :DEFINE command.

### Examples

```

R OPSER )
*IAUTO CTY,ATO

```

To start an automatic startup file.

```

MSGLVL 0

```

An example of an automatic startup file.

```

ITLOG
ISLOG
IDFINE DAE=
DAE=R DAEMON
ISLOG
IDFINE M=
M=R OMOUNT
M=START
ISLOG
IDFINE L=
L=R LPTSPL
L=START
ISLOG
IDFINE B=
B=MJOB 5
B=R BATCON
B=START

```

**PJOB Command**

Function

The PJOB command causes the monitor to respond with the job to which the user's terminal is attached.

Command Format

PJOB

Characteristics

The PJOB command:

Leaves the terminal in monitor mode.

Associated Messages

Refer to Chapter 4.

Example

```
⋮ PJOB ↵  
⋮  
⋮
```



## PLEASE Command

### Function

The PLEASE command allows the user non-conflicting two-way communication with the designated station operator.

### Command Format

PLEASE dev: prog! text

dev = any terminal not assigned to a job (i.e., is not a job's controlling terminal ) with which the user wishes to communicate, including:

- a. TTYn: directs the text to a specific terminal unit. The default is TTY0.
- b. OPRnn: directs the text to the operator's terminal station nn.
- c. (null argument) directs the text to TTY0 at the central station.

prog! = the name of the system program to be run automatically when the message is completed. This argument may appear before or after the device argument and must be concluded with an exclamation point. If PLEASE is entered at the CCL entry point, it reads file nnnPLS.TMP. This file is sent to the designated device. After the operator terminates the request, the specified program will be run at its CCL entry point. Neither the dev: or prog! argument can be used from a Batch control file.

text = the user's message. The argument is required. Characters are not transmitted until the RETURN, vertical tab, or form feed key is depressed, at which point the entire line is transmitted.

When the user depresses the RETURN, vertical tab, or form feed key, a message informing the operator of the caller's station number, proj-prog number or user's name if monitor job tables are available, and text message is printed on dev:. An ESCAPE or control-C on either the user's terminal or dev: causes communication to terminate and the user's TTY to be left in monitor mode. Note that when the line terminates with an ESCAPE, the line is typed but the operator response is not waited for. Messages may be typed in both directions without retying the command.

### Characteristics

The PLEASE command:

Places the terminal in user mode until ESCAPE is typed.

Runs a system program except when used with Batch.

Depends on FTCLX which is normally absent in the DECsystem-1040.

## Restrictions

For Batch users, the PLEASE command is trapped by the Batch Controller and only PLEASE text is allowed. It can be used to request operator action while in the Batch mode. The line of text can only be one line terminated with an ESCAPE.

## Associated Messages

Refer to Chapter 4.

## Example

```
PLEASE TELL ME WHEN DTAS WILL BE FREE )  
OPERATOR HAS BEEN NOTIFIED  
IN HALF AN HOUR  
THANKS  
^C  
  
.
```

## Function

The PLOT command is used to place entries in the plotter output queue. This command is equivalent to the following form of the QUEUE command:

QUEUE PLT:jobname = list of input specifications

## Command Format

PLOT jobname = list of input specifications

jobname = name of the job being entered into the queue. The default is the name of the first file in the request, not the name of the first file given. These differ when the first file given does not yet exist.

input specifications = a single file specification or a string of file specifications, separated by commas, for the disk files being processed. A file specification is in the form dev:file.ext[directory].

dev: = any disk file structure to which PLTSPL will have access; the default is DSK:.

file.ext = names of the files. The filename is optional. The default for the first filename is \*, the default for subsequent files is the last filename used. The extension can be omitted; the default is .PLT.

[directory] = a directory to which the user has access; the user's directory is assumed if none is specified.

The user can obtain the listing of entries in the plotter queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., PLOT [40,15]). If no arguments are given with the command (i.e., only the command name is given), the entries for all jobs of all users are output. The asterisk convention can be used for the input specifications. Switches that aid in constructing the queue entry can appear as part of the input specifications. These switches are divided into three categories:

1. Queue-operation - Only one of these switches can be placed in the command string because they define the type of queue request. The switch used can appear anywhere in the command string.
2. General - Each switch in this category can appear only once in the command string because they affect the entire request. The switch used can appear anywhere in the command string.

3. File control - Any number of these switches can appear in the command string because this category is specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the filename, it becomes the default for subsequent files. For example, the command string

PLOT FILEA,FILEB/DISP:REN,FILEC

indicates that the DISPOSE switch is only for FILEB. The command string

PLOT /DISP:REN FILEA,FILEB,FILEC

indicates that the DISPOSE switch is for all three files.

The following switches can be used with the PLOT command. Note that if an argument to a switch is omitted, the colon preceding the argument must also be omitted. Otherwise, the argument is assumed to be zero and not the default value.

SWITCH	CATEGORY	EXPLANATION
/AFTER:t	General	Process the request after the specified time; t is either in the form of hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting AFTER time must be less than the DEADLINE time. If the switch, or the value of the switch, is omitted, no AFTER constraints are assumed.
/BEFORE:t	General	Queue only the files with creation dates before t where t is in the form dd-mmm-yy hhmm.
/BEGIN:n	File Control	Start the output after n feet. The default is to start output at the beginning.
/COPIES:n	File Control	Repeat the output the specified number of times. n must be less than 64. If more than 63 copies are needed, two separate requests must be made. If the switch is omitted, single copies are output.
/CREATE	Queue Operation	Make a new entry into the plotter output queue. This switch is the default for the queue-operation switches.

SWITCH	CATEGORY	EXPLANATION
/DEADLINE:tt	General	Process the request before the specified time; tt is either in the form hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting DEADLINE time must be greater than the AFTER time. If the switch, or the value of the switch, is omitted, no DEADLINE constraints are assumed.
/DELETE	File Control	Remove the file from the specified directory immediately. (Same as /DISPOSE:RENAME).
/DISPOSE:DELETE	File Control	Delete the file after it has been plotted.
/DISPOSE:PRESERVE	File Control	Do not delete the file after it has been plotted. This is the default for all files except file with extensions .LST, .TMP, and, if the protection is 0xx, .PLT.
/DISPOSE:RENAME	File Control	Remove the file from the specified directory immediately. The DELETE option of the DISPOSE switch does not delete the file until it has been plotted. Thus, it continues to occupy space in the specified directory. The RENAME option frees the space immediately. This is the default for files with extensions .LST, .TMP, and if the protection is 0xx, .PLT.
/F	Queue Operation	List the entries in the plotter queue, but do not update the queues. Therefore, the list may not be an up-to-date listing but the listing will be faster than with /LIST.
/FORMS:a	General	Place the output on the specified form. The argument to the switch must be six alphanumeric characters. The default is that normal forms are used.
/KILL	Queue Operation	Remove the specified entry from the plotter queue. This switch can be used for deleting a previously submitted request as long as the request has not been started by the spoolers.

SWITCH	CATEGORY	EXPLANATION
/LIMIT:n	General	Limit the output to the specified number of pages.
/LIST	General	List the entries in the plotter queue; if the switch, along with all other switches, is omitted, all entries for all jobs of all users are listed.
/MODIFY	Queue Operation	Alter the specified parameters in the job. This switch requires that the user have access rights to the job. It can be used for altering a previously submitted request as long as the request has not been started by the spoolers.
/NEW	File Control	Accept the request even if the file does not yet exist.
/NOTE:a	File Control	Plot the specified text (a) in the output.
/NULL	General	Do not output an error message if there are no files in the request and do not create a queue entry. This is assumed at KJOB time.
/OKNONE	File Control	Do not output message if no files match the wildcard construction. This is assumed at KJOB time. However, a totally null queue request produces a fatal error message.
/PHYSICAL	File Control	Suppress logical device name assignments for the device specified.
/PLOT:ASCII	File Control	Plot the file in ASCII mode. If the /PLOT: switch is omitted, the file is plotted in the data mode specified in the file.
/PLOT:BINARY	File Control	Plot the file in binary mode. If the /PLOT: switch is omitted, the file is plotted in the data mode specified in the file.
/PLOT:IMAGE	File Control	Plot the file in image mode. If /PLOT: switch is omitted, the file is plotted in the data mode specified in the file.

SWITCH	CATEGORY	EXPLANATION
/PRIORITY:n	General	Assign the specified external priority (n=0 to 62) to the request. The larger the number, the greater priority the job has. The default is 10 if no switch is given and 20 if the switch is specified without a value.
/PROTECT:nnn	General	Assign the protection nnn (octal) to the job. If the switch or the value of the switch is omitted, the standard protection is assumed.
/REMOVE	File Control	Remove the file from the queue. This switch is valid only with /MODIFY and can be used to remove a previously submitted file as long as the spoolers have not started processing the request.
/SEQ:n	General	Specify a sequence number to help in identifying a request to be modified or deleted.
/SINCE:t	General	Queue only the files with creation dates after the specified time t where t is in the form dd-mmm-yy hhmm.
/START:n	File Control	Start on the nth line of the file. If the switch, or the value of the switch is omitted, the first line is assumed.
/STRS	File Control	Search for the file on all file structures in the search list and take each occurrence. The default is to take just the first occurrence.
/UNPRESERVED	General	Output the files only if they are not preserved (i.e., the first digit of the protection code is 0). This switch avoids redundant plotting.

#### Characteristics

The PLOT command:

Leaves the terminal in monitor mode.

Runs the QUEUE program, thereby destroying the user's core image.

Does not require LOGIN when only QUEUE listings are desired.

Depends on FTQCOM which is normally absent in the DECsystem-1040.

#### Associated Messages

Refer to Chapter 4.

#### Examples

**\_PLOT \*.PLT/FORMS:PLAIN)**

Cause all files with the extension PLT in the user's area to be plotted. Because these are spooled files (i.e., have the extension .PLT), the files are renamed out of the user's area immediately, and deleted after plotting. The operator is asked to put PLAIN paper on the plotter.



## PRESERVE Command (1)

### Function

The PRESERVE command renames the specified files with the standard protection (usually 055 or 057) inclusively ORed with 100. The files are then preserved and KJOB will not delete them unless requested to. This command has the same action as the P argument to the KJOB command when individually determining what to do with each file.

### Command Format

```
PRESERVE file1.ext,file2.ext,file3.ext,...
```

The full wildcard construction can be used for either the filename or the extension. If a protection code is given, the result of the PRESERVE command is the same as the PROTECT command; that is, the given protection code is used instead of the standard protection inclusively ORed with 100.

### Characteristics

The PRESERVE command:

- Leaves the terminal in monitor mode.

- Runs the PIP program, thereby destroying the user's core image.

- Depends on FTCCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Example

```
.PRESERVE TEST,MAC )  
.PRE PROG, COLE,F4,NAME,* )
```

---

1

This command runs the COMPIL program, which interprets the command before running PIP.

## PRINT Command

### Function

The PRINT command is used to place entries into the line printer output queue. This command is equivalent to the following form of the QUEUE command:

```
QUEUE LPT:jobname = list of input specifications
```

### Command Format

```
PRINT jobname = list of input specifications
```

jobname = name of the job being entered into the queue. The default is the name of the first file in the request, not the name of the first file given. These differ when the first file given does not yet exist.

input specifications = a single file specification or a string of file specifications, separated by commas, for the disk files being processed. A file specification is in the form dev:file.ext [directory].

dev: = any disk file structure to which LPTSPL will have access; the default is DSK:.

file.ext = names of the files. The filename is optional. The default for the first filename is \*, the default for subsequent files in the last filename used. The extension can be omitted; the default is .LPT.

[directory] = a directory to which the user has access; the user's directory is assumed if none is specified.

The user can obtain the listing of entries in the line printer queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., PRINT [10, 157]. If no arguments are given with the command (i.e., only the command name is given), all entries in the line printer queue for all jobs for all users are output.

The asterisk convention can be used for the input specifications. Switches that aid in constructing the queue entry can appear as part of the input specifications. These switches are divided into three categories:

1. Queue-operation - Only one of these switches can be placed in the command string because they define the type of queue request. The switch used can appear anywhere in the command string.
2. General - Each switch in this category can appear only once in the command string because they affect the entire request. The switch used can appear anywhere in the command string.

3. File control - Any number of these switches can appear in the command string because this category is specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the filename, it becomes the default for subsequent files. For example, the command string

PRINT FILEA, FILEB/DISP:REN, FILEC

indicates that the DISPOSE switch is only for FILEB. The command string

PRINT /DISP:REN FILEA, FILEB, FILEC

indicates that the DISPOSE switch applies to all three files.

The following switches can be used with the PRINT command. Note that if an argument to a switch is omitted, the colon preceding the argument must also be omitted. Otherwise, the argument is assumed to be zero and not the default value.

SWITCH	CATEGORY	EXPLANATION
/AFTER:tt	General	Process the request after the specified time; tt is either in the form of hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting AFTER time must be less than the DEADLINE time. If the switch, or the value of the switch, is omitted, no AFTER constraints are assumed.
/BEFORE:t	General	Queue only the files with a creation date before time t, where t is in the form dd-mmm-yy hhmm. If this switch is omitted, no BEFORE constraints are assumed.
/BEGIN:n	File Control	Start the output on the nth page. The default is to begin output on the first page.
/COPIES:n	File Control	Repeat the output the specified number of times. n must be less than 64. If more than 63 copies are needed, two separate requests must be made. If this switch is omitted, one copy is given.
/CREATE	Queue Operation	Make a new entry into the line printer output queue. This switch is the default for the queue-operation switches.

SWITCH	CATEGORY	EXPLANATION
/DEADLINE:tt	General	Process the request before the specified time; tt is either in the form hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting DEADLINE time must be greater than the AFTER time. If the switch, or the value of the switch, is omitted, no DEADLINE constraints are assumed.
/DELETE	File Control	Remove the file immediately from the specified directory. (Same as /DISPOSE:RENAME.)
/DISPOSE:DELETE	File Control	Delete the file after it has been printed.
/DISPOSE:PRESERVE	File Control	Do not delete the file after it has been printed. This is the default for all files except files with extensions of .LST, .TMP, and, if the protection is 0xx, .LPT.
/DISPOSE:RENAME	File Control	Remove the file from the specified directory immediately. The DELETE option of the DISPOSE switch does not delete a file until it has been printed. Thus, it continues to occupy space in the specified directory. The RENAME option frees the space immediately. This is the default for files with extensions .LST, .TMP, and, if the protection is 0xx, .LPT.
/F	Queue Operation	List the entries in the line printer queue, but do not update the queues. Therefore, the list may not be an up-to-date listing but the listing will be faster than with /LIST.
/FILE:ASCII	File Control	Indicate that the input file format is to be interpreted as ASCII text. This is assumed for all files with extensions other than .DAT.
/FILE:COBOL	File Control	Indicate that the input file format is to be interpreted as COBOL SIXBIT text.
/FILE:FORTTRAN	File Control	Indicate that the input file format is to be interpreted FORTRAN ASCII text (obeys FORTRAN carriage control characters). This is assumed if the user explicitly specifies a .DAT extension.

SWITCH	CATEGORY	EXPLANATION
/FORMS:a	General	Place the output on the specified form. The argument to the switch must be six alphanumeric characters. The default is that normal forms are used.
/HEADER:0 or 1	File Control	Output block headers at the beginning of the file, if 1 (default). Do not output headers, if 0.
/KILL	Queue Operation	Remove the specified entry from the Batch input queue. This switch can be used for deleting a previously submitted request as long as the request has not been started by the spooler.
/LIMIT:n	General	Limit the output to the specified number of pages.
/LIST	Queue Operation	List the entries in the line printer queue; if the switch, along with all other switches, is omitted, all entries for all jobs of all users are listed.
/LOG	File Control	Define the file that the spoolers will use to record their process. The default is jobname .LOG.
/MODIFY	Queue Operation	Alter the specified parameters in the job. This switch requires that the user have access rights to the job. It can be used for altering a previously submitted request as long as the request has not been started by the spooler.
/NEW	File Control	Accept the request even if the file does not yet exist.
/NOTE:a	File Control	Print the specified text (a) in the output.
/NULL	General	Do not output an error message if there are no files in the request and do not create a queue entry. This is assumed at KJOB time.
/OKBINARY	File Control	Print files whose extensions imply binary information. Normally files with extensions .SAV, .SHR, .LOW, .REL, and .HGH will not appear in the print queue.

SWITCH	CATEGORY	EXPLANATION
/OKNONE	File Control	Do not output message if no files match the wildcard construction. This is assumed at KJOB time. However, a totally null queue request produces a fatal error message.
/PHYSICAL	File Control	Suppress logical device name assignments for the device specified.
/PRINT:ARROW	File Control	Convert all control characters to up-arrow format except 011-015 and 020-024. This is the default.
/PRINT:ASCII	File Control	Send the file to the line printer with no changes.
/PRINT:OCTAL	File Control	Perform an octal dump of the file.
/PRINT:SUPPRESS	File Control	Suppress all carriage-control characters except for ASCII code characters LF and CR; this switch implies the use of the /PRINT:ARROW and is equivalent to the operator command to the spooler (SUPPRESS).
/PRIORITY:n	General	Assign the specified external priority (n=0 to 62) to the request. The larger the number, the greater priority the job has. The default is 10 if no switch is given and 20 if the switch is specified without a value.
/PROTECT:nnn	General	Assign the protection nnn (octal) to the job. If the switch, or the value of the switch, is omitted, the standard protection is assumed.
/REMOVE	File Control	Remove the file from the queue. This switch is valid only with /MODIFY and can be used to remove a previously submitted file as long as the spooler has not started processing the request.
/REPORT:code	File Control	Print the specified report within a COBOL report file. Code can be up to 12 characters in length.
/SEQ:n	General	Specify a sequence number to help in identifying a request to be modified or deleted.

SWITCH	CATEGORY	EXPLANATION
/SINCE:t	General	Queue only the files with creation dates after the specified time t where t is in the form dd-mmm-yy hhmm.
/SPACING:DOUBLE	File Control	Double-space the output lines.
/SPACING:SINGLE	File Control	Single-space the output lines. This is the default if no /SPACING switch is used.
/SPACING:TRIPLE	File Control	Triple-space the output lines.
/START:n	File Control	Start on the nth line of the file. If the switch, or the value of the switch, is omitted, the first line is assumed.
/STRS	File Control	Search for the file on all file structures in the search list and take each occurrence. The default is to take just the first occurrence.
/UNPRESERVED	General	Output the files only if they are not preserved (i.e., the first digit of the protection code is 0). This switch avoids redundant printing.

#### Characteristics

The PRINT command:

Leaves the terminal in monitor mode.

Runs the QUEUE program, thereby destroying the user's core image.

Does not require LOGIN when only queue listings are desired.

Depends on FTQCOM which is normally absent in the DECsystem-1040.

#### Associated Messages

Refer to Chapter 4.

Examples

**\_PRINT NOTICE, TXT** )

Print the file DSK:NOTICE.TXT.

**\_PRINT SYSTAT, SCM/DISP:REN/COPI2** )

Print two copies of the file DSK:SYSTAT.SCM from the user's default area. Rename the file out of the user's area immediately and delete it after spooling.

**\_PRINT \*, TXT/HEAD:0/FORMS:2PART** )

Print all files in the user's area which have the extension .TXT. Do not print file headers between the files. Print the files on forms known to the operator as 2PART.

**\_PRINT /SEQ:356/KILL** )

Remove the request with sequence number 356 from the LPT queue. This is accepted only if the spooler has not started processing the request.

**\_PRINT LOADER, SAV/OKBINARY/PRINT: SUPPRESS** )

Print a file known to be a binary file and suppress all carriage control characters except CR and LF.

**\_PRINT PRGMAC, REL/PRINT:OCTAL** )

Print an octal dump of the file PRGMAC.REL.



## PROTECT Command (1)

### Function

The PROTECT command alters the access protection codes associated with the specified files. The action of this command is similar to the R switch in PIP.

The access protection of a file is indicated by three octal digits. Each digit represents a particular class of user. The first digit represents the owner of the file, the second represents users with the same project number of the owner, and the third represents all of the other users. Each number in the three digit code can be one of the following.

- 7 No access privileges
- 6 Execute the file
- 5 Read and execute the file
- 4 Append, read, and execute the file
- 3 Update, append, read, and execute the file
- 2 Write, update, append, read, and execute the file
- 1 Rename, write, update, append, read, and execute the file
- 0 Change protection, rename, write, update, append, read, and execute the file.

The standard protection is normally 057 which means the owner has all privileges (0), users in the owner's project can read and execute the file (5), and all other users cannot access the file (7). However, the system standard may be changed by the individual installations. Note that the owner of a file can alter the file's protection regardless of the existing protection code. Thus, codes 0 and 1 for the owner have an equivalent action. In addition, since the owner can LOOKUP the file to change the protection via a RENAME, codes 7 and 6 are equivalent to code 5 for the owner.

### Command Format

PROTECT file1 <nnn>, file2 <nnn>, file3 <nnn>...

The protection can be specified before the filename in which case it is the default for subsequent files until changed. The full wildcard construction can be used for either the filename or the extension.

If the user has the required privileges, he can change the protection of files not in his directory by specifying the desired directory name. If the directory name is before the filename, it becomes the default for all succeeding files.

If the protection code is omitted, the result of the PROTECT command is the same as the PRESERVE command; that is, the standard protection code inclusively ORed with 100 is given to the file.

1

This command runs the COMPIL program, which interprets the command before running PIP.

## Characteristics

The PROTECT command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCLX which is normally absent in the DECsystem-1040.

## Associated Messages

Refer to Chapter 4.

## Examples

```
.PROTECT FORM,*<157>)  
;PRO MAIN,MAC<123>, <456>EQUIL,CBL, ADD,ALG)
```

## Function

The QUEUE command allows the user to make entries in several system queues - the input queue for the Batch system, and the output spooling queues for the line printer, the card punch, the paper-tape punch, and the plotter. The QUEUE command also provides the means of obtaining listings of the entries in the queues.

## Command Formats

1. QUEUE INP: jobname = control file specification, log file specification

To make an entry in the Batch input queue, INP:.

2. QUEUE output queue name: jobname = list of input specifications

To make an entry in an output spooling queue.

3. QUEUE listing file specifications/LIST = list of queue names

To obtain a listing of the entries in a queue.

4. The following six commands can be substituted for the various formats of the QUEUE command:

- a. CPUNCH jobname = list of input specifications equivalent to QUEUE CDP: jobname = list
- b. PLOT jobname = list of input specifications equivalent to QUEUE PLT: jobname = list
- c. PRINT jobname = list of input specifications equivalent to QUEUE LPT: jobname = list
- d. PUNCH jobname = list of input specifications<sup>1</sup> equivalent to QUEUE PTP: jobname = list
- e. SUBMIT jobname = control file name, log file name equivalent to QUEUE INP: jobname = control file, log file
- f. TPUNCH jobname = list of input specifications equivalent to QUEUE PTP: jobname = list

---

<sup>1</sup>The PUNCH command can be redefined by the installation to be equivalent to QUEUE CDP: jobname = list.

Queue names are taken from the following list:

INP: (I:)	Batch input queue
LPT: (L:)	line printer output queue, default condition
CDP: (C:)	card punch output queue
PTP: (PT:)	paper-tape punch output queue
PLT: (PL:)	plotter output queue

Control file specification is the file specification, plus switches and keyword parameters, for the control file being submitted to the Batch input queue. This file can be on any file structure that the user has access to; the default is DSK:. The filename is required, but the extension can be omitted; the default is .CTL. The asterisk construction is legal for the filename or extension and is the only way a user can submit multiple control files to Batch with a single QUEUE command.

Log file specification is the file specification for the file that is to be used to record action taken during the execution of the control file. This file can be in any directory on any file structure in which the user can write. The default is the same file structure in which the control file resides. If the filename is missing, the log file is given the same name as the control file. If the extension is omitted, it is .LOG.

Jobname is the name of the job being entered into the queue. The default jobname is the name of the first file in the request not the first file given. These names are different when the first file given does not yet exist.

Input specifications are the file specifications for the disk files to be processed, and the various switches and keyword parameters that aid in constructing the queue entry. The files can be on any file structure that the queue processor has access to; the default is DSK:. The files can be in any directory, provided that the user has read-access to them; the default is the user's directory. The filename is optional; the default is \* for the first filename. The default for subsequent filenames is the last filename used. Note that the asterisk construction is legal only in the input specifications. The extension can be omitted because each queue has a default extension for the files to be processed. These default extensions are:

- .CTL - Batch input queue
- .LPT - line printer queue
- .CDP - card punch queue
- .PTP - paper-tape punch queue
- .PLT - plotter queue

The listing file specification is the description of the listing file. The default for the listing file destination is TTY unless a filename is specified. In this case, the listing goes to the disk. The default extension for the queue listing is .LSQ. If no queue names are specified, all queues for all the jobs of all users are listed.

The user can obtain queue listings for specific project-programmer numbers by following the queue name with the desired project-programmer numbers enclosed in square brackets (e.g., QUEUE/LIST = INP: [27,235], [10,47]). The wildcard construction can be used for the project number or programmer number. If no queue names are specified but a project-programmer number is given, all queues

for all jobs of the specified user are listed. For example, `QUEUE/LIST = [27,400]` lists the entries in all queues for the user [27,400].

**Switches** - Three categories of switches are provided. The first category contains the switches that define the operation; the second contains the switches that can appear only once because they affect the entire request; the third contains the switches specific to each file. In general, switches that precede the filename become the default for all succeeding files. This is true also for a device name, an extension, or a directory name that precedes a filename.

**Queue-Operation Switches** - Only one of this type of switch can be placed in a command string, because these switches define the type of queue request. This switch may appear anywhere in the command string.

**General Queue Switches** - Each of these switches can appear only once in a command string. They affect the entire request, generally in terms of scheduling. These switches can appear anywhere in the command string.

**File-Control Switches** - These switches affect the individual files in a request and must be adjacent to the filename in the command string. In order to change the defaults for the rest of the files, however, these switches must appear before a filename. For example, the command string

```
QUEUE FILEA,FILEB/DISP:REN,FILEC
```

indicates that the `DISPOSE` switch is only for `FILEB`. The command string

```
QUEUE /DISP:REN FILEA,FILEB,FILEC
```

indicates that the `DISPOSE` switch applies to all three files.

In the table of switches below, the following conventions have been used:

**ALL** - Switches that can appear for both the Batch input queue and the output queues.

**INPUT** - Switches that can appear only for the Batch input queue.

**LIST** - Switches that can appear only for the listing file specification.

**OUTPUT** - Switches that can appear only for the output queues.

Note that if an argument to a switch is omitted, the colon preceding the argument must also be omitted. Otherwise the argument is assumed to be zero and not the default value.

SWITCH	CATEGORY	QUEUES	MEANING
/AFTER:t	General	ALL	Process the request after the specified time. t is in the form hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting AFTER time must be less than the DEADLINE time. If the switch, or the value of the switch, is omitted, no AFTER constraints are assumed.
/BEFORE:t	General	OUTPUT	Queue only the files with creation dates before time t where t = dd-mm-yy hhmm.
/BEGIN	File Control	OUTPUT	Start the output on the nth page, card, or foot. The default is to begin output on the first unit.
/CARDS:n	General	INPUT	Use n (decimal) as the maximum number of cards that can be punched by the job. If the switch is omitted, no cards are punched. If the switch is given with no value, 2000 cards is assumed as the maximum.
/CHARGE:a(1)	General	ALL	Charge the run to the specified account.

---

1

Not yet implemented.

SWITCH	CATEGORY	QUEUES	MEANING
/COPIES:n	File Control	OUTPUT	Repeat the output the specified number of times (n must be less than 64). The default is one copy. If more than 63 copies are desired, two requests must be made.
/CORE:n	General	INPUT	Use n (in decimal K) as the maximum amount of core memory that the job can use. If the switch is omitted, the maximum of 25K is assumed; if the value of the switch is omitted, a maximum of 40K is assumed.
/CREATE	Queue Operation	ALL	Make a new entry in the specified queue. This switch is the default for the queue-operation switches.
/DEADLINE:t	General	ALL	Process the request before the specified time. t is in the form hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting DEADLINE time must be greater than the /AFTER time. If the switch, or the value of the switch is omitted, no DEADLINE constraints are assumed.

SWITCH	CATEGORY	QUEUES	MEANING
/DEFER(1)	Queue Operation	ALL	Make a new entry in the specified queue, but the request is deferred until LOGOUT.
/DELETE	File Control	ALL	Same as /DISPOSE:RENAME.
/DEPEND:n	General	INPUT	Specifies the initial value of the dependency count in decimal. When used with /MODIFY, this switch changes the dependency count of another job. If n is a signed number (+ or-), that number is added to or subtracted from the dependent job's count. If n is not a signed number, the dependent job's count is changed to n. If this switch is omitted, no dependency is assumed.
/DISPOSE:DELETE	File Control	ALL	Delete the file after spooling.
/DISPOSE:PRESERVE	File Control	ALL	Save the file after spooling. This is the default for all files except those with extensions of .LST, .TMP, and if protection is 0xx, .CDP, .LPT, .PLT, .PTP.
/DISPOSE:RENAME	File Control	ALL	Rename the file from the specified directory immediately, remove it from the logged-out quota, and delete it after spooling. This is the default for files

---

1  
Not yet implemented.



SWITCH	CATEGORY	QUEUES	MEANING
			with extensions of .LST, .TMP, and if protection is 0xx, .CDP, .LPT, .PLT, .PTP.
/F	Queue Operation	LIST	List the entries in the queue, but do not update the queues. Therefore, the list may not be an up-to-date listing of the queues but the listing will be faster than with /LIST.
/FEET:n	General	INPUT	Use n (in decimal) as the maximum number of feet of paper tape that the job can punch. If the switch is omitted, no paper tape is punched. If the value is omitted, the default is 10*B+20 feet, where B is the number of blocks in the request.
/FILE:ASCII	File Control	OUTPUT	Specify that the input file format is to be interpreted as ASCII text. This is assumed for all files with extensions other than .DAT.
/FILE:COBOL	File Control	OUTPUT	Specify that the input file format is to be interpreted as COBOL SIXBIT text.
/FILE:ELEVEN	File Control	OUTPUT	Specify that the input file format is to be interpreted as four 8-bit bytes in each 36-bit word. These bytes are arranged as follows: 1st byte is in bits 10-17; 2nd byte, in bits 2-9; 3rd byte, in bits 28-35; 4th byte, in bits 20-27.

SWITCH	CATEGORY	QUEUES	MEANING
/FILE:FORTRAN	File Control	OUTPUT	Specify that the input file format is to be interpreted as FORTRAN ASCII text (obeys FORTRAN control characters). This is assumed if the user specifies the extension .DAT explicitly.
/FORMS:a	General	OUTPUT	Place the output on the named forms. The argument to the switch must be six alphanumeric characters. Normal forms (14 x 11) are used if this switch is omitted. Narrow forms are 8-1/2 x 11.
/HEADER:0 or 1	File Control	OUTPUT	Output block headers at beginning of the file if 1 (default); do not output headers if 0.
/HELP			Print a message giving the general format of the command string and explains the dialogue that is entered if the user needs additional help.
/KILL	Queue Operation	ALL	Remove the specified entry from the specified queue. This switch requires an output specification; it does not default to LPT:*. The /KILL switch can be used for deleting a previously submitted request as long as the request has not been started.

SWITCH	CATEGORY	QUEUES	MEANING
/LIMIT:n	General	OUTPUT	Limit the output to the specified number of pages, cards, feet, or minutes.
/LIST	Queue Operation	LIST	List the specified entries in the queue; the default entries are those for queues for all the jobs of all users.
/LOG	File Control	OUTPUT (LPT)	Define the file that the spoolers will use to record their output. The default is jobname. LOG.
/MODIFY	Queue Operation	ALL	Alter the specified parameters in the specified jobs; this switch requires that the user have access rights to the job. It also requires a queue name; it does not default to the LPT. This switch can be used to modify a previously submitted request as long as the request has not been started.
/NEW	File Control	ALL	Accept request even if file does not yet exist. This is the default for the log file of Batch input queue.
/NOTE:a	File Control	OUTPUT	Output the specified text (a) in the output.
/NULL	General	OUTPUT	Do not output an error message if there are no files in the request and do not create a queue entry. This is assumed at KJOB time.

SWITCH	CATEGORY	QUEUES	MEANING
/OKBINARY	File Control	OUTPUT (LPT)	Print files whose extensions include binary information. Normally files with extensions .SAV, .SHR, .LOW, .REL, and .HGH will not be in print queues.
/OKNONE	File Control	OUTPUT	Do not produce message if no files match the wildcard construction. However, a totally null queue request produces a fatal error message.
/OUTPUT:n	General	INPUT	<p>Cause job to terminate with a /Z:n to KJOB (n is from 0 to 4).</p> <p>N=0 Suppress all normal queuing performed at LOGOUT time.</p> <p>N=1 Queue only the log file.</p> <p>N=2 Queue only the log file and spooled output (e.g., *.LPT).</p> <p>N=3 Queue the log file, spooled output, and *.LST files.</p>

SWITCH	CATEGORY	QUEUES	MEANING
			N=4 Queue the log file, spooled output, *.LST files, and any requests deferred to LOGOUT time (default).
/PAGE:n	General	INPUT	Use n (decimal) as the maximum number of pages of output that the job can print. If the switch is omitted, the maximum is 200 pages; if only the value is omitted, a maximum of 2000 pages can be printed.
/PAPER:x	File Control	OUTPUT	Identical to /PUNCH:x, /PRINT:x, /TAPE:x, or /PLOT:x.
/PHYSICAL	File Control	ALL	Suppress logical device names for the specified device.
/PLOT:ASCII	File Control	OUTPUT (PLT)	Plot the file in ASCII mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.
/PLOT:BINARY	File Control	OUTPUT (PLT)	Plot the file in binary mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.
/PLOT:IMAGE	File Control	OUTPUT (PLT)	Plot the file in image mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.

SWITCH	CATEGORY	QUEUES	MEANING
/PRINT:ARROW	File Control	OUTPUT (LPT)	Convert all control characters to up-arrow format except 011-015 and 020-024. This is the default.
/PRINT:ASCII	File Control	OUTPUT (LPT)	Send the file to the line printer with no changes.
/PRINT:OCTAL	File Control	OUTPUT (LPT)	Print the file in octal.
/PRINT:SUPPRESS	File Control	OUTPUT (LPT)	Suppress all character-control characters except for ASCII code characters LF and CR; this switch implies the use of the /PRINT:ARROW. Equivalent to operator command to spooler (SUPPRESS).
/PRIORITY:n	General	ALL	Give the specified external priority (n = 0 to 62) to the request. A larger number is greater priority. The default is 10 if no switch is given, and 20 if a switch is given without the value.
/PROTECT:nnn	General	ALL	Specify a protection nnn (in octal) for this job or queue entry. If the switch, or the value of the switch, is omitted, the standard protection is assumed.
/PUNCH:026	File Control	OUTPUT (CDP)	Punch files in 026 Hollerith code. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.

SWITCH	CATEGORY	QUEUES	MEANING
/PUNCH:ASCII	File Control	OUTPUT (CDP)	Punch files in ASCII card code. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:BINARY	File Control	OUTPUT (CDP)	Punch files in binary card format. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:D029	File Control	OUTPUT (CDP)	Punch files in the old DEC 029 card code. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:IMAGE	File Control	OUTPUT (CDP)	Punch files in image mode. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/REMOVE	File Control	OUTPUT	Remove the file from the queue. This switch is valid only with the /MODIFY switch and can be used to remove a previously submitted file as long as the Batch System has not started processing the job.
/REPORT:code	File Control	OUTPUT (LPT)	Print the specified report within a COBOL report file. Code can be up to 12 characters in length.

SWITCH	CATEGORY	QUEUES	MEANING
/RESTART:0 or 1	General	INPUT	A value of 0 (default) means the job cannot be requeued or restarted by the operator after a system crash. A message is sent to the job's log file. A value of 1 means the job will be requeued or restarted. The job should not be restartable if there are changes to the permanent file directory.
/SEQ:n	General	ALL	Specify a sequence number to aid in identifying a request to be modified or deleted.
/SINCE:t	General	OUTPUT	Queue only the files with creation dates after the specified time t where t is in the form dd-mmm-yy hhmm.
/SPACING:DOUBLE	File Control	OUTPUT (LPT)	Double-space the output lines.
/SPACING:SINGLE	File Control	OUTPUT (LPT)	Single-space the printed lines (default).
/SPACING:TRIPLE	File Control	OUTPUT (LPT)	Triple-space the printed lines.
/START:n	File Control	ALL	Start on n line of the file. If the switch, or the value of the switch, is omitted, the Batch System starts with the first line.
/STRS	File Control	OUTPUT	Search for the file on all structures in the search list and take each occurrence. The default is to take just the first occurrence of the file.



SWITCH	CATEGORY	QUEUES	MEANING
/TAG:xxx	File Control	INPUT	Start at the statement labelled xxx (up to 5 characters) of the control file. Equivalent to GOTOxxx at the beginning of the control file.
/TAPE:ASCII	File Control	OUTPUT (PTP)	Punch the tape in ASCII code. If the /TAPE switch is not given, the files are punched according to the data mode of the file.
/TAPE:BINARY	File Control	OUTPUT (PTP)	Punch the tape in binary mode. If the /TAPE switch is not given, the files are punched according to the data mode of the file.
/TAPE:IBINARY	File Control	OUTPUT (PTP)	Punch the tape in image-binary mode. If the /TAPE switch is not given, the files are punched according to the data mode of the file.
/TAPE:IMAGE	File Control	OUTPUT (PTP)	Punch the tape in image mode. If the /TAPE switch is not specified, the files are punched according to the data mode of the file.
/TIME:hhmmss	General	INPUT	Specify the central processor time limit for the job. A colon may be used to separate the hours, minutes, and seconds. If no switch is specified, the limit is 5 minutes; if the switch is specified without a value, the limit is 1 hour.

SWITCH	CATEGORY	QUEUES	MEANING
/TPLOT:n	General	INPUT	Use n (decimal minutes) as the maximum amount of plotting time allowed for the job. If the switch is omitted, no plotter time is allowed; if the value is omitted but the switch is given, the maximum plotter time is 10 minutes.
/UNIQUE: 0 or 1	General	INPUT	Run any number of Batch jobs under this project-programmer number at the same time, if 0. Runs only one Batch job at any one time, if 1 (default).
/UNPRESERVED	General	OUTPUT	Output file only if not preserved.
/ZDEFER(1)	Queue Operation	ALL	Create a new entry in a queue and defer it until LOGOUT; however, the deferred file is zeroed first so that all previous /DEFER requests from the current job are deleted.

### Characteristics

The QUEUE command (and its associated variations):

Leaves the terminal in monitor mode.

Runs the QUEUE program, thereby destroying the user's core image.

Does not require LOGIN when only queue listings are desired.

Depends on FTQCOM which is normally absent in the DECsystem-1040.

---

<sup>1</sup>  
Not yet implemented.

## Associated Messages

Refer to Chapter 4.

## Examples

- `._QUEUE FILEA,FILEB)` Enter file FILEA.LPT and FILEB.LPT in the line-printer queue under the jobname of FILEA.
- `._QUEUE INP:=TEST)` Enter file TEST.CTL in the Batch input queue under jobname TEST and log file with name TEST.LOG.
- `._QUEUE IN:PAYR=MAN)` Enter file MAN.CTL in the Batch input queue under jobname PAYR and log file with name MAN.LOG.
- `._QUEUE DSK:A,X=/LIST)` Place a queue listing of all jobs into file A.X in the user's disk area.
- `._QUEUE INP:FREED=FILEA/CREATE)  
/PRIORITY:4/TIME:1:5` Place file FILEA.CTL in the Batch input queue with the jobname FREED. An external priority of 4 and CPU time limit of one minute and five seconds are set for the job. The log file is named FILEA.LOG.
- `._QUEUE INP:TEST=/KILL)` Remove the entry corresponding to TEST.CTL from the Batch input queue.
- `._QUEUE INP:JOBNAM=/MODIFY/TIME:200)` Alter the time parameter of the entry corresponding to JOBNAM.CTL in the Batch input queue.
- `._QUEUE INP:=JOB.CTL/PAGES:500/TPLOT:20)` Establish a limit of 500 pages and 20 minutes of plotting on the output generated by this job.
- `._QUEUE PLT:=JOB.PLT/LIMIT:20)` Queue a file to PLTSPL with a limit of 20 minutes of plotting time.

## QUOLST Program

### Function

The QUOLST program informs the user of both the amount of disk space he has used and the amount he has left on each file structure in his search list. This program also returns the amount of free space that the system has left for all users of the structure. Free system space on structures not in the user's search list is not output. This information can be obtained by typing SYSTAT /F.

The output given for each file structure consists of 1) the structure name, 2) the number of blocks allocated, and 3) the numbers of blocks left in the logged-in quota, in the logged-out quota, and on the structure. The number of blocks allocated is the same as the number output in the summary of the DIRECT command when the /ALLOC switch is used.

Note that the QUOLST program does not return statistics for a user logged-in under [1,2] since this project-programmer number has infinite quotas.

### Command Format

R QUOLST

### Characteristics

The R QUOLST command:

Leaves the terminal in monitor mode.

Runs the QUOLST program, thereby destroying the user's core image.

### Examples

```
.R QUOLST)
USER:    27,400 STR          USED   LEFT:(IN) (OUT) (SYS)
DSKA:     10           1000     100    4703
DSKB:    491           9509     4509   4240
DSKC:      0          10000     5000    396
```

.R QUOLST)

The user is over quota on DSKA: and must delete files before he can logout.

USER:	31,50			
STR	USED	LEFT:(IN)	(OUT)	(SYS)
DSKA:	1022	-22	-922	4215
DSKB:	1735	78265	8265	36
DSKC:	0	2000	1000	6378

## R Command

### Function

The R command loads a core image from the system device and starts it at the location specified within the file (.JBSA). It is equivalent to RUN SYS: file.ext core and is the usual way to run a system program that does not have a direct monitor command to run it.

This command clears all of user core. However, programs should not count on this action and should explicitly clear those areas of core that are expected to contain zeroes (i.e., programs should be self-initializing). This action allows programs to be restarted by a ^C, START sequence without having to do another R command.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (2 EOFs in a row).

### Command Format

R file.ext core

Arguments are the same as in the RUN command except that SYS: is used as the default device. (In nondisk monitors, the default is the generic name that matches the system device.) Refer to the RUN command for a discussion of the core argument.

The extension applies to the low file, not the high file. An extension of .SHR, then .HGH, is assumed for the high file. If the user types an extension of .SHR or .HGH, the extension is treated as a null extension since .SHR and .HGH are confusing as low file extensions.

### Characteristics

The R command:

- Places the terminal in user mode.

- Runs a system program, thereby destroying the user's core image.

### Associated Messages

Refer to Chapter 4.

Examples

```

R PIP )
*
R PIP 5 )
*

```

## REASSIGN Command

### Function

The REASSIGN command allows one job to pass a device to a second job without having the device go through the monitor device pool (restricted or unrestricted). Both restricted and unrestricted devices can be reassigned. This command, applied to DECTapes, clears the copy of the directory currently in core, forcing the next directory reference to read a new copy from the tape, but does not clear the logical name assignment. If a device is INITed, a RELEASE UUC is performed unless the user issuing the command is reassigned the device to himself.

### Command Format

REASSIGN dev job

dev = the physical or logical name of the device to be reassigned. This argument is required.

job = the number of the job to which the device is to be reassigned. If no job is specified, the device is reassigned to the job issuing the command. This is useful when the user wants to force the next directory reference to come from the tape instead of core.

A logical name which is also a physical name can be reassigned only if the job issuing the command and the job to which the device is to be reassigned have the same project-programmer number, or the user issuing the command has operator privileges (logged-in under [1,2] or logged-in at OPR). However, a logical name cannot be duplicated; i.e., two devices cannot have the same logical name.

### Characteristics

The REASSIGN command:

Leaves the terminal in monitor mode.

Requires core.

Does not operate when the device is currently transmitting data.

### Restrictions

The job's controlling terminal cannot be reassigned.



## Associated Messages

Refer to Chapter 4.

## Examples

<code>._REASSIGN LPT:17)</code>	Reassign the line printer to job 17.
<code>._</code>	
<code>._REASSIGN CDP:4)</code>	Reassign the card punch to job 4.
<code>._</code>	
<code>._</code>	

## REATTA Program

### Function

The REATTA program allows a user to transfer his job from one terminal to another. Unlike the ATTACH command, REATTA does not require a password or that the terminal be of the same type that LOGIN recognizes in order to run the job. For example, usually a [1,2] job can run only on a local terminal. However, the REATTA program can be used to attach a [1,2] job from a local terminal to a remote terminal.

Before reattaching his job, the user should verify that the terminal to which he is attaching is turned on and working properly. Otherwise, it might be difficult to retrieve the job.

### Command Format

.R REATTA

REATTA responds by asking for the new terminal name.

TYPE NEW TTY NAME:

The user answers with either the new terminal name (e.g., CTY, TTY2) OR NUMBER (E.G., 2). REATTA then responds with

FROM JOB n

on the old terminal, and

NOW ATTACHED TO JOB n

on the new terminal.

### Characteristics

The R REATTA command:

Leaves the terminal in monitor mode.

Runs the REATTA program, thereby destroying the user's core image.

### Restrictions

The R REATTA command is not available to Batch users.

### Associated Messages

Refer to Chapter 4.

### Examples

```
┆R REATTA )  
TYPE NEW TTY NAME: TTY27 )  
FROM JOB 7 ;appears on old terminal  
┆  
NOW ATTACHED TO JOB 7 ;appears on TTY 27  
┆
```

## REENTER Command

### Function

The REENTER command restarts a program at a location pre-specified in the program as an alternate entry point. Although the use of this alternate entry point varies among programs, it is frequently used for a partial reinitialization of the program. (The START command is used for a complete reinitialization and the CONTINUE command is used for no reinitialization.) Note that the DDT command resembles the REENTER command because it also restarts the program at an alternate entry point. However, this point is fixed since it is the beginning address of DDT.

The REENTER command copies the saved program counter value from .JBPC into .JBOPC and starts the program at an alternate entry point specified in .JBREN (must be set by the user or his program). If the job was executing a UWO when it was interrupted (i.e., in exec mode but not in TTY input wait or SLEEP mode), the monitor continues the job until the UWO is completed and then traps to the REENTER address in .JBREN. If the job is in TTY input wait or SLEEP mode, the trap to the REENTER address occurs immediately and .JBOPC contains the address of the UWO. If the job is in user mode, the trap also occurs immediately. Therefore, it is always possible to continue the interrupted program after trapping by executing a JRSTF@.JBOPC.

### Command Format

REENTER

### Characteristics

The REENTER command:

Places the terminal in user mode.

Requires core.

Requires the user to have a job number.

### Associated Messages

Refer to Chapter 4.

### Example

```
._PEE )
```

## RENAME Command (1)

### Function

The RENAME command changes one or more items of the file specification of files on disk or DECTape.

### Command Format

RENAME arg

arg = a pair of file specifications separated by an equal sign, or a string of such pairs separated by commas:

RENAME new1 = old1,new2 = old2,...

If the new filename is specified without an extension, the null extension is assumed. Device or file structure names can be specified only with the new filename and remain in effect until changed or until the end of the command string is reached. In addition, a protection may be specified with the new filename and remains in effect only for that filename. This command accepts the full wildcard construction.

### Characteristics

The RENAME command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

### Associated Messages

Refer to Chapter 4.

---

1

This command runs the COMPIL program, which interprets the command before running PIP.

Example

```
.RENAME T11.MAC=T1.MAC )  
FILES RENAMED:  
T1.MAC
```

```
.RENAME *.BAC=*.MAC )  
FILES RENAMED:  
T11.MAC  
T2.MAC  
T3.MAC
```

```
.RENAME TEST.MAC<057>=TEXT.MAC )  
FILES RENAMED:  
TFST.MAC
```

.

## RESOURCES Command

### Function

The **RESOURCES** command prints the names of all available devices (except TTY's and PTY's), all file structures, and all physical units not in file structures (unless they are down or nonexistent).

### Command Format

**RESOURCES**

### Characteristics

The **RESOURCES** command:

Leaves the terminal in monitor mode.

Does not require LOGIN.

### Example

```
.RES )  
DSKA,DSKB,DSKC,DPR0,DPB1,CDR0,2,PTR0,LPT0,1,2,3,DTA0,3,4,5,6,  
7,MTAG,1,2,PTP0,CDP0,PLT0,DIS0
```

.

## REWIND Command

### Function

The REWIND command rewinds a magnetic tape or a DECtape. This command is equivalent to the PIP command string:

```
dev: (MW)_
```

### Command Format

REWIND dev:

dev: = a magnetic tape (MTAn) or a DECtape (DTAn).

### Characteristics

The REWIND command:

Leaves the terminal in monitor mode.

Runs the COMPIL program, thereby destroying the user's core image.

Depends on FTCCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4

### Examples

```
._REW DTA4: )
```

```
._REWIND MTA1: )
```



## Function

The RUN command loads a core image from a retrievable storage device and starts at the location specified within the file (.JBSA).

If the program has two segments, both the low and high segments are set up. If the high file has extension .SHR (as opposed to .HGH), the high segment will be shared. Therefore, if the user has RUN (or GET) the same program, I/O will not usually be required for the high segment. A two-segment program may have a low file extension (.LOW).

The RUN command clears all of user core. However, programs should not count on this action and should explicitly clear those areas of core that are expected to contain zeroes (i.e., the programs should be self-initializing). This action allows programs to be restarted by a ^C, START sequence without having to do another RUN command.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (2 EOFs in a row).

## Command Format

RUN dev:file.ext [directory] core

dev: = the logical or physical name of the device containing the core image. The default device name is DSK:. (In nondisk monitors, the default is the generic name that matches the system device.)

file.ext = the name of the file containing the core image; .ext applies to the low file, not the high file. An extension of .SHR, then .HGH, is assumed for the high file. If the user types an extension of .SHR or .HGH, the extension is treated as a null extension since .SHR and .HGH are confusing as low file extensions. The default filename is the job's current name as set by the last R, RUN, GET, SAVE, or SSAVE command, the last SETNAM UUO, or the last command which ran a program.

[directory] = the directory name; required only if core image file is located in a disk area other than the user's.

core = the amount of core to be assigned to the sum of the low and high segments if different from minimum core needed to load the program or from the core argument of the SAVE command which saved the file.

If core < the minimum low segment size, then an error message occurs.

If core > the minimum low segment size and < the sum of the high segment and the minimum low segment size, then the core assignment is the low segment size.

If core > the sum of the minimum low segment and the high segment size, then the core assignment is the size of both the low and high segments to be used.

Core arguments can be specified in units of 1024 words or 512 words (a page) by following the number with K or P, respectively. For example, 2P represents 2 pages or 1024 words. If K or P is not specified, K (1024 words) is assumed.

Note that on KA10 based systems (DECsystem-1040, 1050, 1055), the minimum unit of allocation is 1024 words. Therefore, all arguments are rounded up to the nearest multiple of 1024 words (e.g., 3P is treated as 2K on a KA10 based system).

Since previous core is returned, MTA must have the core argument because there is no directory telling how much core is for the low segment. Refer to Appendix D.

## Characteristics

The RUN command:

Places the terminal in user mode.

## Restrictions

On systems with a large amount of core memory, the user should not specify a core argument that forces the high segment to start higher than 400000 (i.e., a core argument of greater than 128K) unless the program's high segment is location independent. If this is done, the ILLEGAL UUO error message is likely to occur.

## Associated Messages

Refer to Chapter 4.

## Examples

```
._RUN TEST )  
._RUN HISTST [10,63] )  
._RUN DTA3:TFST1 )
```

## SAVE Command

### Function

The SAVE command writes out a core image of the user's core area on the specified device. It saves any user program (two-segment sharable, one-segment nonsharable, or two-segment nonsharable) as one or two files. Later, when the program is loaded by a GET, R, or RUN command, it will be nonsharable. If DDT was loaded with the program, the entire core area is written; if not, the area starting from zero up through the program break (as specified by .JBFF) is written. The save command stores JOBVER in .RBVER when saving .LOW and .SAV files; and it stores JOBHVR in .RBVER when saving .HGH and .SHR files. Refer to DECsystem-10 Monitor Calls for a description of the job data area locations referenced by this command.

The SAVE command should be used instead of the SSAVE command when debugging a two-segment program. Refer to Appendix D for additional information on the SAVE command.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (2 EOFs in a row).

When running under a virtual memory system, DAEMON will be called on a SAVE command if any pages are paged out or the job has a non-contiguous core image.

### Command Format

SAVE dev:file.ext [proj,prog] core

dev = the device on which the core image file is to be written. The default device name is DSK:. In nondisk monitors, the default is the generic name that matches the system device. The colon following the device name is required if a device is specified.

file.ext = the name to be assigned to the core image file. The default filename is the job's current name as set by the last R,RUN GET, SAVE, or SSAVE command, the last command which ran a program (e.g., DIRECT), or the last SETNAM UUU.

ext applies to the file within the low segment, not within the high segment.

[proj,prog] = the name of the disk area on which the core image file is to be written.

core = the amount of core in which the program is to be run. This value is stored in JOBDAT as the job's core area (.JBCOR) and is used by subsequent RUN and GET commands. This argument is optional.

Core arguments can be specified in units of 1024 words or 512 words (a page) by following the number with K or P respectively. For example, 2P represents 2 pages or 1024 words. If K or P is not specified, K (1024 words) is assumed.

Note that on KA10 based systems (DECsystem-1040, 1050, 1055), the minimum unit of allocation is 1024 words. Therefore, all arguments are rounded up to the nearest multiple of 1024 words (e.g., 3P is treated as 2K on a KA10 based system).

If core is omitted, only the number of blocks required by the core image area (as explained in the RUN command description) is assumed.

### Characteristics

The SAVE command:

Leaves the terminal in monitor mode.

Requires core.

Does not operate when a device is currently transmitting data.

### Associated Messages

Refer to Chapter 4.

### Example

```
.SAVE )  
JOB SAVED
```

```
.SAVE DTA3:TEST )  
JOB SAVED
```

## SCHED Command

### Function

The SCHED command types out the schedule bits as set by the operator. This information allows the user to determine the use of the system (e.g., regular timesharing or batch jobs only) before he logs in. The schedule bits are as follows:

- 0 regular timesharing.
- 1 no further logins allowed except from CTY.
- 2 no further logins from remote terminals, and no answering of data sets.
- 4 batch jobs only.
- 100 device mounts can be done without operator intervention.
- 200 unspooling allowed.
- 400 no operator coverage.

### Command Format

SCHED

### Characteristics

The SCHED command:

Leaves the terminal in monitor mode.

Does not require LOGIN.

Depends on output from the SET SCHED command which is normally absent in the DECsystem-1040.

### Example

```
._SCHED )  
0000400
```

Regular timesharing, but no operator coverage.

```
._SCHED )  
0000003
```

No logins allowed from local or remote terminals and data sets are not answered.

```
._SCHED )  
000500
```

Regular timesharing, but no operator coverage. Device mounts can be done without operator interventions.

# SEND Command

## Function

The SEND command provides a mechanism for one-way interconsole communication. (This command replaces the TALK command.) A line of information is transmitted from one terminal to another, with the identification of the terminal sending the information. With remote communications capabilities, SEND is able to differentiate between stations.

When the SEND command is sent from the central station operator's terminal (OPR) or from a terminal logged in as [1,2], it allows a broadcast of a line of information to all non-slaved terminals (including remote terminals) in the system. This allows important information to be dispersed, such as system shutdown or hardware problems. SEND ALL messages do not go to slaved terminals unless the SET TTY NO GAG bit is set to permit reception when the terminal is busy.

A busy test is made on single-destination messages before the message is sent unless the sender or the receiver of the message is OPR or a job logged-in as [1,2]. The receiver of the message is considered busy if his terminal is not at monitor command level. If the receiver is busy, the sender receives the message BUSY and the information is not sent, unless the receiving terminal has the TTY NO GAG bit set (refer to the SET TTY command). If the receiving terminal is turned off, the information appears to have been sent, since the hardware cannot detect this condition on hard-wired terminals.

## Command Format

SEND dev:text

or

SEND JOB n text

dev = any physical terminal name (CTY included) or OPRnn. If OPRnn is specified, the message is sent to the operator at station nn. If OPR (nn is null) is specified, the message is sent to the operator at the user's logical station. If the terminal sending the message is the operator's terminal, the argument may be ALL to provide the broadcast operation.

n = the job number to which the message is to be sent.

the message printed on the receiving terminal appears as follows:

;;TTYn: - text

where

n is the TTY sending the message, and text is the message. A bell sounds on the receiving terminal when the message is sent.

## Characteristics

The SEND command:

Leaves the terminal in monitor mode.

Does not require LOGIN.

## Restrictions

The SEND command is not available to the Batch user.

## Associated Messages

Refer to Chapter 4.

## Examples

```
.SEND OPR: PLEASE WRITE-ENABLE DTA3 )  
!
```

## SET BLOCKSIZE Command

### Function

The SET BLOCKSIZE command sets a default blocksize (in words) for the specified magnetic tape.

### Command Format

SET BLOCKSIZE dev:nnnn

dev: = MTAn: where n is the number of the magnetic tape drive for which the blocksize is to be set, or a logical name associated with a physical magnetic tape. The user must have the magnetic tape assigned to him. This argument is required.

nnnn = a decimal number between 3 and 4094 designating the block size for this magnetic tape. No additional checking is done for the legality of the specified number besides the check for the maximum 4094 and a minimum of 3. This argument is required.

### Characteristics

The SET BLOCKSIZE command:

Leaves the terminal in monitor mode.

Depends on both FTSET and FTMTSET which are normally absent in the DECsystem-1040.

### Examples

```
.SET BLOCKSIZE MTA2:3956 )  
.  
.ASSIGN MTA4:NAME: )  
MTA4 ASSIGNED  
.SET BLOCKSIZE NAME:2000 )  
.
```



## SET BREAK Command

### Function

The SET BREAK command is used (KI10 processors only) primarily during the debugging process. It is primarily useful when the program which is being debugged:

- will not fail when DDT has been loaded.
- destroys DDT when DDT is loaded.
- destroys the contents of a memory location at an unpredictable point during program execution.

It is possible to break when the specified location is read from, written into, and/or fetched. It is also possible to break on monitor references to items in the user's address space. This is useful when the monitor is storing or retrieving arguments to/from unexpected locations in the user's address space because of malformed UUC argument lists.

If the user is breaking on a WRITE condition, the write condition causing the break will not yet have been executed. Therefore, the instruction located at PC and all operands should be examined before continuing program execution.

### Command Format

SET BREAK AT adr ON condition

SET BREAK NO condition

SET BREAK NONE

where: AT and ON are optional portions of the command line.

adr is an octal number in the range 0-777777, representing a user virtual address.

condition is one or more reason for allowing the break to occur.

Multiple conditions may be specified within one command; these conditions are separated from one another by a comma. The possible conditions which may be specified are:

CONDITION	MEANING
READ	Break if the contents of adr are read by the program. Note that this condition causes a break to occur on a read-modify write as well as on a read.
WRITE	Break if the location specified by adr is written into.
EXECUTE	Break if an instruction is fetched from the location specified by adr.
ALL	Break if the location specified by adr is read from (READ), written into (WRITE), or fetched from (EXECUTE).
MUOU	Break on monitor references as qualified by READ, WRITE, and/or EXECUTE.

If all three break conditions are to be specified, the user can include the word ALL in the command line, replacing the command argument string: WRITE, READ, EXECUTE. When desiring breaks on monitor references for all three break conditions, the user can include one of the following command argument strings in the command line: either MUOU, ALL or MUOU, READ, WRITE, EXECUTE.

When issuing a SET BREAK command without specifying any break conditions, the conditions specified in the previous SET BREAK command are still in effect. If there was no previous SET BREAK command, ALL is assumed. When issuing a SET BREAK command without specifying any address, the conditions included in the command line are OR'ed with existing break conditions and the previously specified address is used. If there are no existing break conditions, 0 is the default address.

In summary, break addresses remain in effect until changed; and break conditions remain in effect until removed.

If the user wishes to remove a break condition, the condition to be removed may be specified in the following command:

```
SET BREAK NO condition
```

If all existing break conditions are to be removed, the following command line may be issued:

```
SET BREAK NONE
```

This command will remove all existent break conditions, but will not remove a previously specified address. An example setting a default break location is:

```
SET BREAK 1000 ON READ, WRITE, EXECUTE
SET BREAK NO READ
SET BREAK NONE
SET BREAK EXECUTE ;1000 IS DEFAULT BREAK LOCATION
```

When a break occurs, one of the messages:

```
%ADDRESS BREAK AT USER PC xxxxxx
```

```
%ADDRESS BREAK AT EXEC xxxxxx UO AT USER xxxxxx
```

will be typed; and the terminal will be left in monitor mode. The second message is produced when MUUO was included in the SET BREAK command line. If the user types:

```
.CONTINUE
```

the program will continue execution at the instruction which caused the break.

#### Characteristics

Requires LOGIN.  
Leaves the terminal in monitor mode.  
K110 processors only.

#### Error Messages

```
?NOT AVAILABLE
```

This message is typed on the user's terminal if the operator has restricted the use of this feature, or if the console switches are set to disallow the use of programmable address break.

#### Examples

```
.SET BREAK AT 1000 ON READ, WRITE )  
.  
.SET BREAK EXECUTE )  
.  
.SET BREAK NONE )  
.  
.SET BREAK 1000 MUUO, EXECUTE, WRITE )
```

## SET CDR Command

### Function

The SET CDR command sets the filename for the next card-reader spooling intercept (refer to DECsystem-10 Monitor Calls). This command is generally not needed, even when the card reader is being simulated on the disk via the spooling mechanism. It is included in case the user wishes to reset or change the spooling. In addition, the Batch Controller uses this command to read spooled input card decks.

### Command Format

SET CDR filename

filename = one- to three-character filename to be used on next card-reader INIT.

### Characteristics

The SET CDR command:

Leaves the terminal in monitor mode.

Depends on FTSET and FTSPLE which are normally absent in the DECsystem-1040.

### Examples

```
._SET CDR A )  
._SET CDR MAS )  
._
```

## SET CPU Command

### Function

The SET CPU command allows a privileged user to change the CPUs on which his job can run. If is used in a multiprocessing system to specify whether the programs run under the job can be processed on the primary CPU, the secondary CPU, or either CPU. The job remains with the specified CPU until (1) another SET CPU command with a different specification is given, (2) a KJOB command is issued, or (3) the user's program overrides the SET CPU command by issuing the SETUOO with a different specification. If the SETUOO overrides the command, the specification given in the UOO remains in effect until a RESET or EXIT UOO or another SETUOO with a different specification is executed. When an EXIT or RESET UOO is executed, the job reverts back to the specification given in the last SET CPU command. When the user logs in, the CPU specification is usually set to ALL. The schedulers for each CPU compete for jobs with the ALL specification so that the load is dynamically balanced between CPUs. Therefore, this command is generally not needed but is provided in case the user wishes to change the CPU specification.

### Command Formats

1. SET CPU CP<sub>xn</sub>

adds the specified CPU to the job's CPU specification.

2. SET CPU NO CP<sub>xn</sub>

removes the specified CPU from the job's CPU specification.

3. SET CPU ALL

adds all of the CPUs to the job's CPU specification.

4. SET CPU ONLY CP<sub>xn</sub>

changes the CPU specification so that it includes only the specified CPU.

x = either U designating a logical name or A or I designating physical names for a KA10 processor (DECsystem-1055) or a K110 processor (DECsystem-1077), respectively.

n = a decimal number from 0 to the number of processors in the system.

## Characteristics

The SET CPU command:

Leaves the terminal in monitor mode.

Depends on FTSET and FTMS which are normally absent in the DECsystem-1040, 1050, and 1070.

## Restrictions

The privileges required for using this command are determined by bit 5 (JP.CCC) of the privilege word, .GTPRV.

## Associated Messages

Refer to Chapter 4.

## Examples

```
.SET CPU ONLY CPU1 )  
.SET CPU CPA0 )  
.
```

## SET DENSITY Command

### Function

The SET DENSITY command sets a default density for the specified magnetic tape.

### Command Format

SET DENSITY dev:nnn

dev: = MTAn: where n is the number of the magnetic tape drive for which the density is to be set, or a logical name associated with a physical magnetic tape. The user must have the device assigned to him. This argument is required.

nnn = 200 bpi  
556 bpi  
800 bpi

This argument is required.

### Characteristics

The SET DENSITY command:

Leaves the terminal in monitor mode.

Depends on both FTSET and FTMTSET which are normally absent in the DECsystem-1040.

### Examples

```
._SET DENSITY MTA5: 556 )  
._
```

## SET DSKFUL Command

### Function

The SET DSKFUL command controls the treatment of a user's job when it is attempting output and either there is no space available on the file structure being referenced or the user's quota for that structure is exceeded.

### Command Formats

#### 1. SET DSKFUL ERROR

Output stops and an error condition is passed to the program. Most programs respond to the error condition by issuing an error message and returning the job to monitor level without any opportunity for the user to continue.

#### 2. SET DSKFUL PAUSE

Output stops and execution of the program is suspended. An error message is printed on the user's terminal and control of the job is returned to the monitor. Generally, the user should employ the SEND command to request assistance from the operator. Execution of the program can be resumed with the CONTINUE command as long as the user does not give a command which destroys the core image of the interrupted program. However, the program will again be stopped if the problem of insufficient disk space or insufficient quota has not been corrected in the interim.

The default setting is ERROR unless the user's accounting file entry specifies PAUSE as his default.

### Characteristics

The SET DSKFUL command:

- Leaves the terminal in monitor mode.

- Depends on FTSET which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Examples

```
._SET DSKFUL PAUSE ↵
```



## SET DSKPRI Command

### Function

The SET DSKPRI command allows a privileged user to set the priority for his job's disk operations (data transfers and head positionings). The standard priority is 0, and the range of permissible values is -3 to +3. This means that a priority lower than the standard can be specified, as well as higher than the standard. The priority specified applies to all disk I/O channels currently open or subsequently opened whose priority has not been explicitly set with a DISK. UUO (refer to DECSYSTEM-10 Monitor Calls). The priority specified in the SET DSKPRI command remains in effect until (1) another SET DSKPRI command is given with a different priority, (2) a KJOB command is issued, or (3) the user's program overrides the SET DSKPRI command by issuing a DISK. UUO with a different priority.

### Command Format

SET DSKPRI n

n = a decimal number from -3 to +3 indicating the priority to be associated with the job's disk operations. When n = 0, the priority is the normal timesharing priority.

### Characteristics

The SET DSKPRI command:

Leaves the terminal in monitor mode.

Depends on both FTSET and FTDPRI which are normally absent in the DECSYSTEM-1040.

### Restrictions

The privileges required for using this command are determined by bits 1 and 2 of the privilege word, .GTPRV. These two bits specify an octal number from 0-3. The user is always allowed a 0 priority.

### Examples

```
.SET DSKPRI 2 )  
.
```

## SET HPQ Command

### Function

The SET HPQ command allows a privileged user to place his job in a high-priority scheduler run queue. With this command, the user obtains a faster response and CPU time than in the normal timesharing queues. The job remains in the specified high-priority queue until (1) another SET HPQ command to a different high-priority queue is given, (2) a KJOB command is issued, or (3) the user's program overrides the SET HPQ command by issuing an HPQ UUC with a different value. If an HPQ UUC overrides the command, the level specified in the UUC remains in effect until a RESET or EXIT UUC or another HPQ UUC with a different value is executed. When an EXIT or RESET UUC is executed, the job is returned to the high-priority queue specified in the SET HPQ command.

### Command Format

SET HPQ n

n = a decimal number from 0 to 15 indicating the high-priority queue to be entered. When n = 0, the queue is the normal timesharing run queue. Queue numbers from 1 to 15 are high-priority queues. The number of high-priority queues is an installation parameter and may be less than 15.

### Characteristics

The SET HPQ command:

Leaves the terminal in monitor mode.

Depends on both FTSET and FTHPQ which are normally absent in the DECsystem-1040.

### Restrictions

The privileges required for using this command are determined by bits 6 through 9 of the privilege word, .GTPRV. These four bits specify an octal number from 0-17, which is the highest priority queue attainable by the user.

Associated Messages

Refer to Chapter 4.

Examples

```
.SET HPQ 4 )  
:
```

## SET PHYSICAL LIMIT Command

### Function

The SET PHYSICAL LIMIT command is used to specify the maximum current physical page limit (CPPL), by including the word LIMIT within the command line. (CPPL is described in Chapter 1 of the *Monitor Calls Manual*). By including the word GUIDELINE in the command line, the SET PHYSICAL command is used to establish a guideline for the page fault handler. The page fault handler will then use the specified figure as a guideline in determining the exact time a program will go virtual.

### Command format

$$\text{SET PHYSICAL } \left\{ \begin{array}{l} \text{LIMIT} \\ \text{GUIDELINE} \end{array} \right\} \left\{ \begin{array}{l} \text{nP} \\ \text{nK} \end{array} \right\}$$

where: LIMIT and GUIDELINE are alternative portions of the command line. If both are omitted, GUIDELINE is assumed.

1K equals 1024 words, and 1P equals a page of 512 words. If K and P are omitted, K is assumed.

K may be specified within the range 1 to 256K; P may be specified within the range 1 to 512P.

If the command SET PHYSICAL LIMIT is given with a 0 argument, the job will never “go virtual”.

### Characteristics

Requires LOGIN.

Leaves the terminal in monitor mode.

### Associated Messages

Refer to Chapter 4.

Example

```
._SET PHYSICAL 100P )  
._SET PHYSICAL GUIDELINE 2K )  
._SET PHYSICAL LIMIT 50K )  
._
```

## SET SPOOL Command

### Function

The SET SPOOL command adds devices to or deletes devices from the current list of devices being spooled for this job. Spooling is the mechanism by which I/O to or from slow-speed devices is simulated on disk. Devices capable of being spooled are: the line printer, the card punch, the card reader, the paper tape punch, and the plotter.

### Command Formats

1. SET SPOOL dev1,dev2,...devn  
adds the specified devices to the job's spool list.
2. SET SPOOL ALL  
places all spooling devices into the spool list.
3. SET SPOOL NONE  
clears the entire spool list.
4. SET SPOOL NO dev1,dev2,...devn  
removes the specified devices from the job's spool list.  
  
dev1, dev2, ...devn = physical device names of one or more devices to be added to or deleted from the current spool list. These names are taken from the following list: CDP, CDR, LPT, PLT, PTP.

### Characteristics

The SET SPOOL command:

Leaves the terminal in monitor mode.

Depends on both FTSET and FTSP which are normally absent in the DECsystem-1040.

### Restrictions

To unspool devices, the job must have (1) the privilege bit set in .GTDRV, (2) bit 28 (200 octal) set in the STATES word by the operator SET SCHED command, or (3) the user must be logged-in under [1,2].

## Associated Messages

Refer to Chapter 4.

## Examples

```
⋮SET SPOOL CDP: )  
⋮SET SPOOL NO LPT: )  
⋮SET SPOOL NONE )  
⋮
```

## SETSRC Program

### Function

The SETSRC program is used to manipulate the job's search list or the system's search list. A search list is defined to be the order of file structures that are to be searched whenever generic device DSK: is explicitly or implicitly specified by the user. This search list is originally defined by the system manager to include the file structures which the user can access. With the SETSRC program, the user can alter the search list defined for him by adding or deleting file structures.

The search list is in the form

fs1/s/s, fs2/s/s, ..., FENCE, ....., fs9/s/s

where fs is the name of the file structure and /s is a switch modifying the file structure. The file structures on the left of the FENCE comprise the active search list and represent the generic device DSK for this job. The files to the right of the FENCE comprise the passive search list and represent file structures that were once in the active search list. File structures are kept in the passive search list in order that quotas can be checked on a DISMOUNT or KJOB command. The FENCE represents the boundary between the active and passive search list.

Note that the MOUNT and DISMOUNT commands can also change the job's search list by adding or deleting a file structure. Since the SETSRC program does not create a UFD if one does not exist, the MOUNT command should be used to create a UFD. The name of the new file structure is placed at the end of the search list.

Refer to the SETSRC specification in the DECsystem-10 Software Notebooks for a complete description of the SETSRC program.

### Command Format

#### R SETSRC

The user can then respond with any of the following commands:

Command	Explanation
A	Add one or more file structures to the existing search list. The file structures (with any switches) are appended to the beginning or the end of the active search list according to the following specifications: <ol style="list-style-type: none"><li>1. If no asterisk appears in the specification (e.g., fs1, fs2) or if an asterisk appears before the file structure names (e.g., *, fs1, fs2), the file structures are added to the end of the search list.</li></ol>



Command	Explanation
	<p>2. If the asterisk follows the file structure names (e.g., fs1, fs2, *), the file structures are added to the beginning of the search list.</p> <p>3. If the asterisk appears in the middle of the file structures (e.g., fs1, *, fs2), the file structures before the asterisk are added to the beginning of the search list and the file structures after the asterisk are added to the end.</p>
	<p>If the specified file structure is currently in the search list, it is removed and then added in the desired position. Therefore, this command can be used to reorder the search list.</p>
C	Create a new search list for this job. Any file structures in the current search list which are not in the new list are moved to the passive search list.
CP	Create a new default directory path.
CS	Create a new system search list (i.e., the file structure search list for device SYS:). The user must be logged in under [1,2] to use this command.
H	Obtain information about the available commands.
M	Modify the current search list and DSK specification by altering the switch settings for individual file structures. This command does not add or remove file structures from the search list.
R	Remove file structures from the search list. They are placed on the right side of the FENCE (passive search list) so that on subsequent LOGOUTs or DISMOUNTs quota limits can be checked.
T	Type the search list of the job.
TP	Type the default directory path.
TS	Type the system search list.

The following switches can be used in the SETSRC command string. Switches that modify file structures must appear immediately after the file structure that they modify. Other switches can appear anywhere in the command string. The switches can be abbreviated as long as the abbreviation is unique. The minimum number of characters is underlined below.

Switches that modify file structures.

/CREATE Allow new files to be created on the file structure.

/NOCREATE Do not allow new files to be created on the file structure when DSK is specified, but allow files to be superseded. Files can be created on the file structure if the user specifies the file structure name explicitly.

`/NOWRITE` Do not allow writing on the file structure for this job (i.e., the file structure is read only).

`/WRITE` Allow writing on the file structure.

If no switches are specified, `/CREATE` and `/WRITE` are assumed. For compatibility with previous versions of SETSRC, `/N` is equivalent to `/NOCREATE` and `/R` equivalent to `/NOWRITE`.

Switches that modify the directory path (used only with the CP command)

These switches can be typed in directly as commands by omitting the CP command and the slash (i.e., `/SCAN` is equivalent to `CP/SCAN`).

`/NOSCAN` Cancel the scan switch for the directory path.

`/SCAN` Set the scan switch for the directory path.

Switches that modify the DSK or SYS specification (used only with the C and M commands)

These switches can be typed in directly as commands by omitting the C or M command and the slash (i.e., `NOSYS` is equivalent to `M/NOSYS`).

`/LIB:` [proj,prog] Set the job's library directory to the UFD [proj, prog] and add it to the user's DSK specification. This means that if a file is not found in the user's directories in his search list, the library directory will then be searched for the file.

`/NOLIB` Remove the library directory from the user's DSK specification.

`/NOSYS` Remove the SYS specification from the user's DSK specification.

`/NONEW` Remove the [1,5] directory from the user's SYS specification.

`/SYS` Add the SYS specification to the user's DSK specification. This means that if a file cannot be found in the user's directories in his search list or in his library directory (if `/LIB:` [proj,prog] has been specified), the system directory [1,4] will then be searched for the file.

`/NEW` Add the directory [1,5] to the user's SYS specification. This means that when the system directory is searched, the directory [1,5] will be searched before the directory [1,4].

## Characteristics

### The R SETSRC command

Places the terminal in user mode.

Runs the SETSRC program, thereby destroying the user's core image.

## Restrictions

The user must be logged in under [1,2] to create a new system search list. The directory path commands (CP and TP) are meaningful only with the 5.04 and later monitors and only if FTSFD is on.

## Examples

```
_R SETSRC )
```

```
*1 )
```

```
DSKB: , FENCE
```

The user's search list is defined as DSKB.

```
*A DSKA: )
```

Add DSKA to the end of the search list.

```
*T )
```

```
DSKB: , DSKA: , FENCE
```

The user's search list is defined as DSKB, DSKA.

```
*A DSKC: , * )
```

Add DSKC to the beginning of the search list.

```
*T )
```

```
DSKC: , DSKB: , DSKA: , FENCE
```

Remove DSKA from the search list.

```
*R DSKA: )
```

```
*T )
```

```
DSKC: , DSKB: , FENCE , DSKA:
```

The user's search list is defined as DSKC, DSKB.

```
*M DSKB: /NOWRITE )
```

Do not allow writing on DSKB.

```
*M /LIB: [27,500] )
```

Set the user's library directory to [27,500] and add it to the user's DSK specification.

```
*SYS )
```

Add SYS: to the user's DSK specification.

```
*T )
```

```
/LIB: [27,500] /SYS DSKC: , DSKB: /NOWRITE , FENCE , DSKA:
```

The user's DSK and SYS specifications are listed first followed by the user's search list.

```
*TS )
```

```
DSKA: , DSKB: , DSKC: , FENCE
```

The system search list is defined as DSKA, DSKB, DSKC.

## SET TIME Command

### Function

The SET TIME command sets a central processor time limit for a job. When the time limit is reached, the job is stopped and a message is typed. A timesharing job may be continued by typing CONT, but no time limit is in effect unless it is reset. A Batch job cannot be continued.

### Command Format

SET TIME n

n = number of seconds of central processor time to which the job is limited. An argument of 0 cancels the time remaining.

### Characteristics

The SET TIME command:

Leaves the terminal in monitor mode.

Depends on both FTSET and FTTLIM which are normally absent in the DECsystem-1040.

### Restrictions

The SET TIME command is ignored in a Batch control file. A Batch job has its time limit set via the /TIME switch in the QUEUE or SUBMIT command string or on the \$JOB card.

### Associated Messages

Refer to Chapter 4.

## Examples

```
.MAKE LOOP.F4)
```

Create a program with an infinite loop.

```
*I10    CONTINUE  
        GOTO 10  
        END
```

```
SS  
*FXSS
```

```
.TYPE LOOP.F4)  
10      CONTINUE  
        GOTO 10  
        END
```

Type the program.

```
.LOAD LOOP)  
FORTRAN: LOOP.F4  
LOADING
```

Compile and load the program.

```
LOOP 2K CORE
```

```
EXIT  
.SET TIME 5)
```

Set the time limit to 5 seconds.

```
.TIME)
```

Clear the incremental run time, so that the SET TIME command can be checked.

```
3.50  
5.57  
KILO-CORE-SEC=32
```

```
.START)
```

Start the loop.

```
?  
?TIME LIMIT EXCEEDED
```

As expected, the time limit was exceeded.

```
.TIME)  
5.00  
10.57  
KILO-CORE-SEC=67
```

```
:
```

## SET TTY or TTY Command

### Function

The SET TTY command (or TTY command) declares properties of the terminal on which the command is typed to the scanner service. With terminals connected directly to the DECsystem-10, the system manager can set some default conditions, so that this command is usually not needed. However, the user is likely to need this command when connected by telephone lines to a dial-up data set, since it may not be possible for the system manager to predict the type of terminal which will be used.

### Command Formats

1. SET TTY NO word

equivalent to TTY NO word

2. SET TTY word

equivalent to TTY word

NO = the argument that determines whether a bit is to be set or cleared. This argument is optional.

word = the various words representing bits that may be modified by this command. The words are as follows:

SET TTY ALTMODE      Converts the ALTmode codes of 175 and 176 to the ASCII standard escape character 033 (initial state).

SET TTY NO ALTMODE   Restores the individual identity of the codes 175 and 176.

SET TTY BLANKS       Restores multiple carriage return-line feeds and form feeds (initial state).

SET TTY NO BLANKS    Suppresses blank lines (consecutive carriage return-line feeds after the first) and outputs form feeds and vertical tabs as 2 carriage return-line feeds. This is useful for a video display terminal in order to increase the amount of output which fits on the screen.

SET TTY CRLF         Restores the carriage return (initial state).

SET TTY NO CRLF      The carriage return normally output at the end of a line exceeding the carriage width is suppressed.

SET TTY DEBREAK      Notifies the system that the terminal has a feature that allows the computer to lock the keyboard. (Model 2741 terminals).

SET TTY NO DEBREAK   Turns off the SET TTY DEBREAK command.

SET TTY ECHO           Restores the normal echoing of each character typed in. Most terminals on the DECsystem-10 are used in full-duplex mode which means that the terminal does not print each character as the user types it. Instead the monitor must echo each character typed so that the character will be printed. If a terminal prints ^C when it is typed but does not print any of the non-control characters typed by the user, then this command can be used to establish normal echoing. (Initial setting determined by each installation when constructing its monitor at MONGEN time.)

SET TTY NO ECHO       This command suppresses monitor echoing of input characters. Local copy terminals (i.e., terminals which automatically print each character as the user types it) do not require normal monitor echoing, since the echoing would cause each character typed to be printed twice. This command is used to stop double-printing of each character typed on a local copy terminal. (Initial setting determined by each installation when constructing its monitor at MONGEN time.)

SET TTY ELEMENT n    Changes the typing element number for Model 2741-type terminals. The element numbers available are:

ELEMENT	Meaning
987	APL correspondence
029	Standard correspondence
087	Call 360 BASIC
963	Extended binary
938	BCD
988	APL (EBCD)

Note that the SET TTY ELEMENT n command also sets the terminal as if the SET TTY NO LC command were used.

SET TTY NO ELEMENT   Turns off the SET TTY ELEMENT n command.

SET TTY FILL n

The filler class n is assigned to this terminal. Many different types of terminals are supported by the Decsystem-10 operating system. Some terminals require one or more filler characters to be sent following certain control characters such as line feed (LF) and horizontal tab (HT). The table below illustrates the number of fillers sent for each character and filler class. The filler characters are CR (215 octal for even parity) for carriage return characters (CR, 215 octal) and DEL (RUBOUT, 377 octal) for all other characters. No fillers are supplied for image mode output (initial setting determined by each installation when constructing its monitor at MONGEN time).

SET TTY NO FILL

Equivalent to TTY FILL 0 (initial setting determined by each installation when constructing its monitor at MONGEN time). Fillers for output and echoing are determined from the following:

Character Name	Octal	Number of Fillers for Filler Class			
		0	1	2	3
BS	010	0	2	6	6
HT	011	0	1 or 2	0	1 or 2(1)
LF	012	0	2	6	6
VT	013	0	2	6	6
FF	014	0	12	21	21
CR on output	015	0	1	3	3(2)
free CR(3)		0	2	4	4
CRLF on output	015-012	0	3	9	9(4)
XON	021	1	1	1	1
TAPE	022	1	1	1	1
XOFF	023	1	1	1	1
NTAP	024	1	1	1	1

(1)1 if 0-3 spaces to tab stop; 2 if 4-7 spaces to tab stop.  
 (2)Output only; no fillers on input.  
 (3)Refer to the SET TTY CRLF command.  
 (4)Sum of the fillers output for a CR and LF.

SET TTY FORM

This terminal has hardware FORM (PAGE) and VT (vertical tab) characters (initial setting determined by each installation when constructing its monitor at MONGEN time).



SET TTY NO FORM	The monitor sends eight line feeds for a FORM and four line feeds for a VT (initial setting determined by each installation when constructing its monitor at MONGEN time).
SET TTY GAG	Messages transmitted by the SEND command cannot be received at this terminal unless the terminal is at command level (initial state).
SET TTY NO GAG	Messages transmitted by the SEND command can be received at this terminal even though it is not at command level.
SET TTY LC	The translation of lower-case character input to upper case is suppressed.
SET TTY NO LC	This command causes the monitor to translate lower-case characters to upper case as they are received. Some terminals have only upper-case characters, while others have both upper and lower case. Frequently, it is convenient to have a terminal with both upper and lower case simulate the behavior of one with upper-case only. This command causes the monitor to perform this simulation. The echo sent back by the monitor always matches the case of the characters after translation. By looking at the printout, the user can determine what translation was performed by the monitor (initial state).
SET TTY PAGE	This command gives the user the ability to temporarily suspend system timeout without losing it. The XOFF key (^S) suspends the timeout, and the XON key (_Q) restores it. The XOFF and XON keys are not echoed and are not sent to the user's program. This command is useful for video display terminals where the user may want to read a page of text before it disappears from the screen. Note that this preempts the use of ^S and ^Q for reading paper tape (see SET TTY TAPE).
SET TTY NO PAGE	The timeout control ability of the XOFF and XON keys is disabled. The current interpretation of these keys depends on the last SET TTY TAPE command (initial state).
SET TTY RTCOMPATIBILITY	Disables the ↑R and ↑T features.

SET TTY NO RTCOMPATIBILITY	Turns on the ↑R and ↑T features.
SET TTY SLAVE	The terminal becomes slaved, i.e., no commands may be typed on the terminal, and the terminal may be ASSIGNED by another user. The user can slave his own terminal and must contact the operator in order to unslave it (initial setting is determined by each installation when constructing its monitor at MONGEN time).
SET TTY TAB	This terminal has hardware TAB stops every eight columns (initial setting is determined by each installation when constructing its monitor at MONGEN time).
SET TTY NO TAB	The monitor simulates TAB output from programs by sending the necessary number of SPACE characters.
SET TTY TAPE	The XON key (^Q) causes the terminal to read paper tape. The XOFF key (^S) causes the terminal to stop reading paper tape. Refer to Chapter 5 of DECsystem-10 Monitor Calls.
SET TTY NO TAPE	The XON key (^Q) and the XOFF key (^S) have no special paper tape function. They may have a PAGE function (initial state).
SET TTY TIDY	Specifies (for Model 2741 terminals) that every character occupies one print space. The terminal normally types out characters so that they appear the same as the user types them in. For example, [ prints out as ^<. In TIDY mode, [ prints out as <.
SET TTY NO TIDY	Turns off the SET TTY TIDY command.
SET TTY SPEED n	Set to n baud the speed at which the system sends and receives. (The user adjusts the speed at the terminal by setting a switch on the terminal.)
SET TTY SPEED m,n	Set the terminal transmitting speed to n and the receiving speed to m. (This is especially useful for display terminals that permit transmitting and receiving at different speeds.)
SET TTY UC	Equivalent to SET TTY NO LC.
SET TTY NO UC	Equivalent to SET TTY LC.

SET TTY WIDTH n      The carriage width (the point at which a free carriage return is inserted) is set to n. The range of n is 17 (two TAB stops) to 200 decimal. The initial state is 72.

#### Characteristics

The SET TTY command:

Leaves the terminal in monitor mode.

Does not require LOGIN.

Depends on FTSET which is normally absent in the DECsystem-1040. However, the TTY command format can always be used.

#### Restrictions

The SET TTY (or TTY) command is not available to the Batch user.

## SET VIRTUAL LIMIT Command

### Function

The SET VIRTUAL LIMIT command is used to specify the current virtual page limit (CVPL). CVPL is described in Chapter 1 of the *Monitor Calls Manual*.

### Command Format

SET VIRTUAL LIMIT  $\left\{ \begin{array}{l} nK \\ nP \end{array} \right\}$

where: LIMIT is an alternate portion of the command line.

1K equals 1024 words, and 1P equals a page of 512 words. If K and P are omitted, K is assumed.

K may be specified within the range 1 to 256K; P may be specified within the range 1 to 512P.

If the command SET VIRTUAL LIMIT is given with a 0 argument, the value specified for CVPL is used. CVPL is set by the system administrator and indicates the current virtual page limit.

### Characteristics

Requires LOGIN.

Leaves the terminal in monitor mode.

### Associated Messages

Refer to Chapter 4.

### Example

```
.SET VIRTUAL LIMIT 32K )  
:
```

## SET WATCH Command

### Function

The SET WATCH command sets the system to print incremental job statistics automatically. This command provides the user with a tool for measuring the performance of his programs.

### Command formats

1. SET WATCH arg(1),arg(2),...,arg(n)  
prints the specified WATCH statistics.
2. SET WATCH ALL  
prints all the WATCH statistics.
3. SET WATCH NONE  
eliminates the printing of all WATCH statistics.
4. SET WATCH NO arg(1), arg(2),...,arg(n)  
eliminates the printing of the specified WATCH statistics.

The following arguments enable printing whenever a monitor command switches the console from monitor to user mode.

arg = DAY prints the time of day, as [HH:MM.SS]

arg = VERSION prints the version of the program in standard format (refer to the VERSION command).

The following arguments enable printing whenever the console is returned to monitor mode via the ↑C,EXIT,HALT,ERROR IN JOBn, or DEVICE xxx OPR zz ACTION REQUESTED messages.

arg = READ prints the incremental number of disk blocks read modulo 4096.

arg = RUN prints the incremental run time.

arg = WAIT prints the wait time (time elapsed since the user started or continued the program).

arg = WRITE prints the incremental number of disk blocks written modulo 4096.

When an UNLOAD is performed, MTA prints magtape statistics in the following form.

[MTAxm/REELID READ (W/H/S) = a/b/c WRITE (W/H/S) = d/e/f]

where: a = words read  
b = hard read errors  
c = soft read errors  
d = words written  
e = hard write errors  
f = soft write errors

Whenever a=b=c=0, the information on READ will not be printed. Whenever d=e=f=0, the WRITE will not be printed.

This message will be produced by an UNLOAD (MTAP11) UUU.

Any combination of the arguments may be specified in any order. Statistics are not printed for commands that do not run programs, such as ASSIGN or PJOB. When a user logs in, his job is set to WATCH the statistics of which he has notified the system manager. The information on what statistics to WATCH is kept in ACCT.SYS.

The order of the error message is the same as the order of output. Therefore, a user who forgets either the argument or the significance of the statistics can find these out by typing SET WATCH and examining the message. A single space is always typed between each statistic, whether the statistic appears or not; therefore, it is possible to tell which statistics are being typed.

#### NOTE

Enabling WATCH output interacts with the incremental data typed by the TIME and DSK commands.

#### Characteristics

The SET WATCH command:

Leaves the terminal in monitor mode.

#### Associated Messages

Refer to Chapter 4.

Examples

1. .SET WATCH P)  
?ARGS ARE: DAY,RUN,WAIT,READ,WRITE,VERSION,ALL,NONE

.SET WATCH)  
?ARGS ARE: DAY,RUN,WAIT,READ,WRITE,VERSION,ALL,NONE

.SET WATCH DAY RUN WAIT READ WRITE

.R PIP)  
[22:38:19]  
\*°C  
[0.10 2.95 457 243]

2. .SET WATCH VERSION DAY)

.R TECO)  
[9:44:30]  
[S:TECO 22(64) + 1  
\*°C

## SKIP Command (1)

### Function

The SKIP command moves a magnetic tape forward a specified number of files or records or to the logical end of tape. This command, depending on its arguments, is equivalent to the following PIP command strings:

```
MTAn: (M #nA)_  
MTAn: (M #nD)_  
MTAn: (M #nT)_
```

SAVED files on magnetic tape always contain two files, a high segment file and a low segment file. If one of the segments is missing, a null file containing one record is written for the missing segment. Thus, in order to space over a SAVED file, the user must skip two files. Refer to Appendix D for the format of a SAVED file.

### Command Formats

1. SKIP MTAn: x FILES

advances forward x files.

2. SKIP MTAn: x RECORDS

advances forward x records.

3. SKIP MTAn: EOT

advances forward to the logical end of tape.

The words FILES, RECORDS, and EOT can be abbreviated to F, R, and E, respectively.

### Characteristics

The SKIP command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

---

1

This command runs the COMPIL program, which interprets the command before running the PIP program.



## Associated Messages

Refer to Chapter 4.

## Examples

```
._SKIP MTA0: 4 FILES )  
._SKIP MTA1: EOT )  
._SKIP MTA2: 20 RECORDS )
```

## SSAVE Command

### Function

The SSAVE command is the same as the SAVE command except that the high segment, if present, will be sharable when it is loaded with the GET command. To indicate this sharability, the high segment is written with extension .SHR instead of .HGH. A subsequent GET will cause the high segment to be sharable. Because an error message is not given if the program does not have a high segment, a user can use this command to save system programs without having to know which are sharable.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (2 EOFs in a row).

The SAVE command rather than the SSAVE command, should be used when debugging the program. This is because a GET command after a SAVE command does not reinitialize the original high segment from the file after the user modifies it with the D command or the DDT program. Refer to Appendix D for more information on the SAVE command.

When running under a virtual memory system, DAEMON will be called on a SSAVE command if pages are paged out or the core image is non-contiguous.

### Command Format

```
SSAVE dev:file.ext [proj,prog] core
```

Arguments and defaults are the same as in the SAVE command.

### Characteristics

The SSAVE command:

- Leaves the terminal in monitor mode.

- Requires core.

- Does not operate when a device is currently transmitting data.

### Associated Messages

Refer to Chapter 4.

Example

.\_SSAVE DSK:TEST )  
JOB SAVED

.

.\_LOAD FILE1 )  
MACRO: FILE1  
LOADING

Compile and load program.

LOADER 1K CORE  
EXIT

.\_SSAVE )  
JOB SAVED

Save a sharable copy. The filename is taken from the routine that contained the starting address.

.\_GET )  
JOB SETUP

Get a sharable copy.

## START Command

### Function

The START command begins execution of a program either previously loaded with the GET command or interrupted while running (e.g., ^C). The old program counter is copied from .JBPC to .JBOPC. An explicit start address is optional, and, if omitted, the address supplied in the file (.JBSA) is used. If an address argument is specified and the job was executing a UUU when interrupted (i.e., it was in exec mode but not in TTY input wait or SLEEP mode), the monitor sets a status bit (UTRP) and continues the job at the location at which it was interrupted before trapping to the specified START address. When the UUU processing is completed, the monitor clears the status bit, sets .JBOPC to the address following the UUU, and then traps to the START address. If the job is in TTY input wait or SLEEP mode, the trap to the program occurs immediately, and .JBOPC contains the address of the UUU. If the job is in user mode, the trap also occurs immediately.

### Command Format

START adr

adr = the address at which execution is to begin if other than the location specified within the file (.JBSA). This argument is optional. If adr is not specified, the address comes from .JBSA. A starting address of 0 may be specified.

### Characteristics

The START command:

Places the terminal in user mode.

Does not operate when a device is currently transmitting data.

Requires core.

Requires LOGIN if an address argument is specified.

### Associated Messages

Refer to Chapter 4.

### Example

```
._START )
```

## SUBMIT Command

### Function

The SUBMIT command is used to place entries into the input queue for the Batch system. This command is equivalent to the following form of the QUEUE command:

```
QUEUE INP: jobname = control file, log file
```

### Command Format

```
SUBMIT jobname = control file, log file
```

jobname = name of the job being entered into the queue.

control file = name of the control file. This file contains all monitor-level and user-level commands for processing by the Batch Controller (BATCON).

log file = name of the log file. This file is used by the Batch Controller to record its processing of the job.

Only the two files mentioned above can be specified in a request to the Batch input queue. The name of the control file is required; the log file name is optional and, if omitted, is taken from the control file. If the jobname is omitted, it is the name of the first file in the request, not the name of the first file given. If an extension is omitted, the following are assumed:

```
.CTL for the control file  
.LOG for the log file.
```

The user can obtain the listing of entries in the input queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., SUBMIT [27.400]). If no arguments appear in the command string (i.e., only the command name is given), all entries in the batch input queue for all jobs are listed.

Three categories of switches can be used in the command string:

1. Queue-operation - Only one of these switches can be placed in the command string because this category defines the type of queue request. The switch used can appear anywhere in the command string.
2. General - Each switch in this category can appear only once in the command string because these switches affect the entire request. The switch used can appear anywhere in the command string.

3. File control - Any number of these switches can appear in the command string because these switches are specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the filename, it becomes the default for subsequent files. For example, the command string

SUBMIT FILEA/DISP:PRE,FILEB

indicates that the DISPOSE switch applies only to FILEA. The command string

SUBMIT /DISP:PRE FILEA,FILEB

indicates that the switch applies to both files.

The following switches can be used with the SUBMIT command. Note that if an argument to a switch is omitted, the colon preceding the argument must also be omitted. Otherwise the argument is assumed to be zero and not the default value.

SWITCH	CATEGORY	EXPLANATION
/AFTER:tt	General	Process the request after the specified time; tt is either in the form of hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting AFTER time must be less than the DEADLINE time. If the switch, or the value of the switch, is omitted, no AFTER constraints are assumed.
/CARDS:n	General	Use n (decimal) as the maximum number of cards that can be punched by the job. If the switch is omitted, no cards are punched. If the switch is given with no value, 2000 cards is assumed as the default.
/CORE:n	General	Use n (decimal K) as the maximum amount of core memory that the job can use. If the switch is omitted, 25K is the maximum. If the switch is specified, but the value is omitted, the default maximum is 40K.
/CREATE	Queue Operation	Make a new entry into the Batch input queue. This switch is the default for the queue-operation switches.
/DEADLINE:tt	General	Process the request before the specified time; tt is either in the form hhmm (time of day) or +hhmm (time later than the current time). A colon may be

SWITCH	CATEGORY	EXPLANATION
		used to separate the hours from the minutes (e.g., hh:mm). The resulting DEADLINE time must be greater than the AFTER time. If the switch, or the value of the switch, is omitted, no DEADLINE constraints are assumed.
/DELETE	File Control	Remove the file from the specified directory immediately. (Same as /DISPOSE:RENAME.)
/DEPEND:n	General	Specify the initial value of the dependency count (in decimal). When used with /MODIFY, this switch changes the dependency count of another job. If n is a signed number (+ or -), that number is added to or subtracted from the dependent job's count. If n is not a signed number, the dependent job's count is changed to n. If this switch is omitted, no dependency is assumed.
/DISPOSE:DELETE	File Control	Delete the file after processing.
/DISPOSE:PRESERVE	File Control	Do not delete the file after processing. This is the default for all files except those with extensions .TMP, .LST, .CDP, .LPT, .PLT, and .PTP.
/DISPOSE:RENAME	File Control	Remove the file from the specified directory immediately. The DELETE option of the DISPOSE switch does not delete the file until it has been processed. Thus, the file continues to occupy space in the specified directory. The RENAME option frees the space immediately. This is the default for files with extensions .TMP, .LST, .CDP, .LPT, .PLT, and .PTP.
/F	Queue Operation	List the entries in the input queue, but do not update the queues. Therefore, the list may not be an up-to-date listing, but the listing will be faster than with /LIST.
/FEET:n	General	Use n (decimal) as the maximum number of feet of paper tape that the job can punch. If the switch is omitted, no paper tape is punched. If the value is omitted, the default is 10*B+20 feet, where B is the number of blocks in the request.

SWITCH	CATEGORY	EXPLANATION
/KILL	Queue Operation	Remove the specified entry from the Batch input queue. This switch can be used for deleting a previously-submitted request as long as the request has not been started by the Batch Controller.
/LIST	Queue Operation	List the entries in the input queue: if the switch, along with all other switches, is omitted, all entries for all jobs of all users are listed.
/MODIFY	Queue Operation	Alter the specified parameters in the job. This switch requires that the user have access rights to the job. It can be used for altering a previously submitted request as long as the request has not been started by the Batch Controller.
/NEW	File Control	Accept the request although the file does not yet exist. This is the default for the log file. When placing this switch with the control file, the user can submit his job and then create the control file.
/OUTPUT:n	General	<p>Cause job to terminate with a /Z:n to KJOB (n is from 0 to 4).</p> <p>N=0 Suppress all normal queuing performed at LOGOUT time.</p> <p>N=1 Queue only the log file.</p> <p>N=2 Queue only the log file and spooled output (e.g., *.LPT).</p> <p>N=3 Queue the log file, spooled output, *.LST files.</p> <p>N=4 Queue the log file, spooled output, *.LST files, and any requests deferred to LOGOUT time (default).</p>
/PAGE:n	General	Use n (decimal) as the maximum number of pages of output that the job can print. If the entire switch is omitted, the maximum is 200 pages; if only the value is omitted, the maximum is 2000 pages.
/PHYSICAL	File Control	Suppress logical device name assignments for the device specified.



SWITCH	CATEGORY	EXPLANATION
/PRIORITY:n	General	Assign the specified external priority (n=0 to 62) to the request. The larger the number, the greater priority the job has. The default is 10 if no switch is given and 20 if the switch is specified without a value.
/PROTECT:nnn	General	Assign the protection nnn (octal) to the job. If the switch, or the value of the switch, is omitted, the standard protection is assumed.
/RESTART:0 or 1	General	A value of 0 means the job is not queued or restarted by the Batch Controller after a system crash (default). A message is sent to the job's log file. A value of 1 means the job is restarted by the Batch Controller.
/SEQ:n	General	Specify a sequence number to help in identifying a request to be modified or deleted.
/START:n	File Control	Begin on the nth line of the control file. If the switch, or the value of the switch, is omitted, the Batch Controller starts with the first line.
/TAG:xxx	File Control	Start at the statement labelled xxx (up to 5 characters) of the control file. Equivalent to a GOTO xxx at the beginning of the control file.
/TIME:hhmmss	General	Specify the central processor time limit for the job. A colon may be used to separate the hours, minutes, and seconds. If no switch is specified, the limit is 5 minutes; if the switch is given without a value, the limit is 1 hour.
/TPLOT:n	General	Use n (decimal minutes) as the maximum amount of plotting time allowed for the job. If the switch is omitted, no plotter time is allowed; if the value is omitted, but the switch is given, the maximum is 10 minutes.
/UNIQUE:0 or 1	General	Run any number of Batch jobs under this project-programmer number at the same time, if 0. Run only one Batch job at any one time, if 1 (default).

## Characteristics

The SUBMIT command:

Leaves the terminal in monitor mode.

Runs the QUEUE program.

Does not require LOGIN when only queue listings are desired.

Depends on FTQCOM which is normally absent in the DECsystem-1040.

## Associated Messages

Refer to Chapter 4.

## Examples

```
.SUBMIT USRJOB=CONTRL,LOGFIL )
```

The defaults are as follows:

1. control file name is CONTRL.CTL
2. log file name is LOGFIL.LOG
3. no cards punched (/CARDS:0)
4. maximum core of 25K (/CORE:25)
5. no dependency (/DEPEND:0)
6. control and log files are saved after spooling (/DISPOSE: PRESERVE)
7. no paper tape punched (/FEET:0)
8. all line printer output is spooled with the maximum pages being 200 (/OUTPUT:4, /PAGE:200)
9. priority is 10 (/PRIORITY:10)
10. standard protection is assumed (/PROTECT:nnn (standard))
11. job is not restarted after a crash (/RESTART:0)
12. control file is begun on the first line (/START:1)
13. maximum CPU time is 5 minutes (/TIME:0:05)
14. no plotter time allowed (/TPLOT:0)
15. only one job at a time under a given project-programmer number is run (UNIQUE:1)

```
.SUBMIT USRJOB=/MODIFY/FEET:35/CORE )
```

Modify the original request to include 35 feet as the maximum number of feet of paper tape that the job can punch and 40K of core as the maximum amount of core that the job can use. This command is valid only if the job has not been started yet by the Batch system.

```
.SUBMIT USRJOB=/KILL )
```

Kill the job only if it has not been started by the Batch system.

## SYSTAT Command

### Function

The SYSTAT command runs a system program which prints status information about the system. This information allows a user to determine the load on the system before logging-in.

To write the output on the disk as a file with name SYSTAT.TXT, assign device DSK with logical name SYSTAT.

The SYSTAT command types the status of the system: system name, time of day, date, uptime, percent null time (idle plus lost time), number of jobs in use.

It types the status of each job logged-in: job number; project-programmer number (\*\*,\*\* = detached, [OPR] = the project-programmer number of the operator, [SELF] = user's project-programmer number); terminal line number (CTY = console terminal, DET = detached, Pn = PTY number); program name being run; program size; job and swapped state (refer to DECsystem-10 Monitor Calls); run time since logged-in.

It types the status of high segments being used: name (PRIV = nonsharable, OBS = superseded); device or file structure name from which the segment came; directory name (\*\*,\*\* if detached); size (SW = swapped out, SWF = swapped out and fragmented, F = in core and fragmented on disk, SPY = user is executing the SPY UUO); number of user in core or on the disk.

The command types swapping space used, virtual core used, swapping ratio, active swapping ratio, virtual core saved by sharing, average job size.

It types status of busy devices: device name, job number, how device is assigned (AS = ASSIGN command, INIT = INIT or OPEN UUO, AS+INIT = both ways).

It types system file structures: free blocks, mount count, single-access job.

It types remote stations: number of station, status of station.

It types dataset control: number of the TTY, status of TTY.

### Command Format

SYSTAT arg

arg = one or more single letters (in any order) used to type any subset of the SYSTAT output. This argument is optional and if it is omitted, the entire SYSTAT output is listed. The following lists the various arguments to the SYSTAT command.

B	busy device status
D	dormant segment status
E	non-disk error report
F	file structure status
G	other system status
H	help text listing the arguments
J	job status
L	list the SYSTAT output on LPT
N	non-job status (i.e., all information except J)
P	disk performance
R	remote station status
S	short job status
T	dataset status
X	read the file DSK:CRASH.XPN

The user can obtain output for individual jobs by specifying one of the following after the command:

A number n which causes information to be listed only for the indicated job (i.e., job n). A period causes information for the user's job (i.e., the job running SYSTAT) to be output.

A project-programmer number specification enclosed in square brackets causes information to be output only for jobs with the specified project-programmer number. The project and/or programmer number can be wild (i.e., specified with an asterisk).

A number preceded by a number sign (#n) causes information to be output only for jobs from the indicated terminal (i.e., TTYn). In addition, a C following the command indicates CTY. Pnn indicates PTYnn, Tnn indicates TTYnn, and a period indicates the terminal on which the SYSTAT command is issued.

## Characteristics

The SYSTAT command:

Leaves the terminal in monitor mode.

Runs the SYSTAT program, thereby destroying the user's core image.

Does not require LOGIN.

Depends on FTCCCLX which is normally absent in the DECsystem-1040.

Examples

A SYSTAT of the K110. Note that core sizes are given in pages (P).

.SYSTAT

STATUS OF RV7AYC K110 SYS#514 AT 3:35:17 ON 01-OCT-84

UPTIME 12:58, 75%NULL TIME = 74%IDLE + 1%LOST  
 19 JOBS IN USE OUT OF 50. 19 LOGGED IN, 1 DETACHED OUT OF 50. (LOGMAX)

JOB	WHO	LINE#	WHAT	SIZE(P)	STATE	RUN TIME
1	400,3536	7	DIRECT	7+19	TC SW	2
2	[OPR]	1	TSTSUP	12	TC SW	20
3	[OPR]	DET	DAEMON	20+SPY	SL	3
4	37,772	5	COPYED	4+19	TI	7
5	37,2072	33		4	TC SW	0
6	136,2252	16	TECO	5+6	TI	5
7	[OPR]	P0J41	LPTSPL	7+7	HB	1
8	[OPR]	P2J41	BATCON	3+6	HB SW	1
9	12,3560	45	DIRECT	7+19	TC SW	1
10	[OPR]	P3J41	OPRONO	5+12	RN	1
11	[OPR]	P4J41	SPRINT	9+6	TC SW	1
12	156,2143	25	UMOUNT	4+9	SL SW	2 \$
13	400,525	41	PIP	3+9	TI	2
14	[OPR]	P5J41	JUSTIF	14+26	SL	20
15	[OPR]	P6J41	ALCOMP	8+15	SL	6
16	16,107	P7J41	SYSTAT	13+SPY	SL	1 \$
17	[SELF]	36	SYSTAT	12+SPY	RN	1 \$
18	[OPR]	P1J41	LPTSPL	6+7	HB	1
41	[OPR]	CTY	OPSER	2+5	SL	4

\$ MEANS EXECUTE ONLY

JNN IS THE CONTROLLING JOB, PNN CORRESPONDS TO TTY212+NN

HIGH SEGMENTS:

PROGRAM	DEVICE	OWNER	HIGH(P)	USERS
OPSER	DSKB	SYS	5	1
LPTSPL	DSKB	SYS	7	2
BATCON	DSKB	SYS	6 SW	1
OMOUNT	DSKB	SYS	12	1
SPRINT	DSKB	SYS	6 SW	1
ALCOMP	DSKC	SYS	15	1
JUSTIF	DSKC	SYS	26	1
DIRECT	DSKB	SYS	19 SW	2
PIP	DSKB	SYS	9	1
TECO	DSKB	SYS	6	1
COPYED	DSKC	SYS	19	1
UMOUNT	DSKB	SYS	9 SW	1

SWAPPING SPACE USED = 185/1600 = 12%  
 VIRT. CORE USED = 284/1600 = 18%  
 SWAPPING RATIO = 284/232 = 1.2  
 ACTIVE SWAPPING RATIO = 29/232 = 0.1  
 VIRT. CORE SAVED BY SHARING = 26/(26+284) = 8%  
 AVERAGE JOB SIZE = 145/19 = 7.6P+165/19 = 8.7P TOTAL=310/19 = 16.3P

BUSY DEVICES:

DEVICE	JOB	WHY	LOGICAL
LPT0	7	AS	
LPT1	18	AS	
23 DISK DDBS			

SYSTEM FILE STRUCTURES:

NAME	FREE	MOUNT
DSKA	1565	16
DSKN	6355	13
DSKB	107476	21
DSKC	35181	21
TOTAL FREE	150577	

REMOTE STATIONS  
NUMBER STATUS

1	CENTRAL
2	NOT IN CONTACT

DATASET CONTROL  
TTY# STATUS

60	IN USE
63	IN USE

.SYSTAT/P)

STATUS OF RV7AYC K110 SYS#514 AT 10:24:14 ON 01-OCT-84

DISK PERFORMANCE STATISTICS:

UNIT OR F/S	BR	BW	DR	DW	XR	XW	MR	MW
DSKM	2085	FREE						
DPA5(416):	18152229717	FREE, 2211	SEEKS					
	100	29717	303	124	10	1	4660	1201
MSB	ERRORS: 1	DEV:1	RETRIES:1	2CONI:20,15	1CONI:20,20015	LBN:25077	2DAT	
AI:537261,440000	1DATAI:537261,460400							
GRUB	68780	FREE						
DPA1(GRUB0):	18770892693	FREE, 967	SEEKS					
	293	118	3504	34	10	1	1388	168

September 1974

```

MSB
SRJ2      79606 FREE
RPA3(SRJ2): 19351806033 FREE, 35 SEEKS
           0           0           28           0           10           1           96           0
ZMT MSB  ERRORS: 2CONI:4000,200015 1CONI:4000,200015 LBN:54 2DATAI:2000
0,50700 1DATAI:20000,50700
DSKV      28615 FREE
DPA6(DSKV00): 19235413661 FREE, 727 SEEKS
           961           840           774           374           10           1           936           345

MSB
DSKA      1482 FREE
FHA0(0NC745): 13458524379 FREE, 0 SEEKS
           185           9           768           91           10           1           3495           701
ERRORS: 1DEV:1 RETRIES:1 2CONI:64000,15 1CONI:64000,115 LBN:2640 2DATAI
:200 1DATAI:200
FHA1(UNC746): 13486049604 FREE, 0 SEEKS
           0           0           4158           1371           10           1           2319           2082
ERRORS: 1DEV:2 RETRIES:1 2CONI:64000,15 1CONI:64000,115 LBN:1852 2DATAI
:27 1DATAI:105
DSKN      7035 FREE
DPA0(BRP022): 13564430908 FREE, 9099 SEEKS
           3319           1219           19080           257           10           1           8215           2874
MSB  ERRORS: CER:6 LBN:1
DSKB      100976 FREE
RPA0(PRODB0): 13641501734 FREE, 11553 SEEKS
           22304           17166           7343           347           10           1           12782           1808
MSB  ERRORS: 1DEV:1 RETRIES:1 2CONI:4000,15 1CONI:4000,20015 LBN:40767
2DATAI:10700 1DATAI:10700
RPA1(PRODB1): 13731417581 FREE, 5721 SEEKS
           17959           18530           4658           354           10           1           2031           1603
           .
           .
           .
OFL ZMT PNM
DPA4( ): 0 FREE, 0 SEEKS
           0           0           0           0           10           1           0           0
OFL ZMT PNM

```

```

ACTIVE SWAPPING STATISTICS:
UNIT  R      W      USED(P)
FHA0  238648  89748  331/400 = 83%
FHA1  122136  93620  261/400 = 65%
FSA0  66608   56536  314/400 = 79%
FSA1   0       0       0/400 = 0%

```

The disk performance statistics given in the above example are labeled BR, BW, DR, DW, XR, XW, MR and MW. An explanation of these titles follows:

- BR = number of 128-word blocks read in buffered mode.
- BW = number of 128-word blocks written in buffered mode.
- DR = number of 128-word blocks read in dump mode.

- DW = number of 128-word blocks written in dump mode.
- XR = number of 128-word blocks read via extended RIBS
- XW = number of 128-word blocks written via extended RIBS
- MR = number of monitor 128-word blocks read
- MW = number of monitor 128-word blocks written

MR and MW correspond to the number of monitor overhead (SAT, UFD, MFD, BAT) blocks read or written, but they do not include the number of blocks swapped in or out. The number of blocks swapped in or out is given as part of the swapping statistics. The total number of blocks read on any given unit is specified by summing BR, DR, and MR blocks swapped in for that unit. The total number of blocks written is found in the same manner.

The number of seeks given for each unit is equal to the sum of the monitor, user (both buffered and dump) and swap seeks.

.SYSTAT/J

STATUS OF RV7AYC K110 SYS#514 AT 8:39:55 ON 01-OCT-84

UPTIME 17:37, 75%NULL TIME = 74%IDLE + 1%LOST  
 21 JOBS IN USE OUT OF 50. 21 LOGGED IN, 1 DETACHED OUT OF 50. (LOGMAX)

JOB	WHO	LINE#	WHAT	SIZE(P)	STATE	RUN TIME
1	400,3536	7	DIRECT	7+19	↑C SW	2
2	[OPR]	1	DIRECT	6+19	↑C	27
3	[OPR]	DEI	DAEMON	20+SPY	SL SW	6
4	37,772	5	COPYED	6+19	TI	7
5	37,2072	33	SOS	12+15	TI	14
6	136,2252	16	TECO	5+6	TI	7
7	[OPR]	P0J41	LPISPL	7+7	HB SW	1
8	[OPR]	P2J41	BATCON	3+6	HB SW	1
9	12,3560	45	TECO	5+6	↑C	4
10	[OPR]	P3J41	OPROMO	5+12	SL SW	2
11	[OPR]	P4J41	SPRINT	9+6	↑C SW	1
12	156,2143	25	UMOUNT	4+9	↑C SW	3 \$
14	[OPR]	P5J41	JUSTIF	14+26	SL SW	21
15	[OPR]	P6J41	ALCOMP	8+15	SL	3
16	16,107	P7J41	SYSTAT	13+SPY	SL	1 \$
17	[SELF]	36	SYSTAT	12+SPY	RN	2 \$
18	[OPR]	P1J41	LPISPL	6+7	HB SW	1
19	156,3507	27	COPYED	4+19	TI	5
20	66,3024	46	UMOUNT	4+9	↑C	2 \$
21	60,60	63	PIP	4+9	TO	2
41	[OPR]	CTY	OPSER	2+5	HB	4

5 MEANS EXECUTE ONLY

JWN IS THE CONTROLLING JOB, PWN CORRESPONDS TO TTY212+NN



HIGH SEGMENTS:  
PROGRAM DEVICE OWNER HIGH(P) USERS

OPSER	DSKB	SYS	5	1
LPTSPL	DSKB	SYS	7 SW	2
BATCON	DSKB	SYS	6 SW	1
OMOUNT	DSKB	SYS	12 SW	1
SPRINT	DSKB	SYS	6 SW	1
ALCOMP	DSKC	SYS	15	1
JUSTIF	DSKC	SYS	26 SW	1
DIRECT	DSKB	SYS	19	2
PIP	DSKB	SYS	9	1
TECO	DSKB	SYS	6	2
SOS	DSKB	SYS	15	1
COPYED	DSKC	SYS	19	2
UMOUNT	DSKB	SYS	9	2

SWAPPING SPACE USED = 229/1600 = 14%

VIRT. CORE USED = 310/1600 = 19%

SWAPPING RATIO = 310/232 = 1.3

ACTIVE SWAPPING RATIO = 12/232 = 0.1

VIRT. CORE SAVED BY SHARING = 60/(60+310) = 16%

AVERAGE JOB SIZE = 156/21 = 7.4P+214/21 = 10.2P TOTAL=370/21 = 17.6P

## TECO Command (1)

### Function

The TECO command opens an already existing file on disk for editing with TECO. Refer to the TECO manual in the DECsystem-10 Software Notebooks.

### Command Format

TECO dev:file.ext[directory]

dev: = the device or file structure name containing the existing file. If omitted, DSK: is assumed.

file.ext = the filename and filename extension of the existing file. If omitted, the arguments of the last EDIT-class command are used.

[directory] = the directory name in which the file appears. If omitted, the user's directory is assumed.

Switches can be passed to TECO by preceding the switch with a slash in the TECO command string. When COMPIL interprets the command string, it passes the switches on to TECO.

### Characteristics

The TECO command:

Places the terminal in user mode.

Runs the TECO program, thereby destroying the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

---

1

This command runs the COMPIL program, which interprets the commands before running TECO.

Example

```
._TECO TEST1.MAC )  
_*^C  
._TECO DSKB;FILNAM,CBL [100,27] )
```

## TIME Command

### Function

The TIME command causes typeout of the total running time since the last TIME command, followed by the total running time used by the job since it was initialized (logged-in), followed by the integrated product of running time and core size (KILO-CORE-SEC=). Time is typed in the following format:

hh:mm:ss.hh

where

hh = hours

mm = minutes

ss.hh = seconds to nearest hundredth.

Interrupt level and job scheduling times are charged to the user who was running when the interrupt or rescheduling occurred.

### NOTE

If automatic runtime is enabled using the SET WATCH command, the incremental runtime is usually 0.

### Command Format

TIME job

job = the job number of the job whose timing is desired. If job is omitted, the job to which the terminal is attached is assumed. In this case, monitor types out the incremental running time (running time since last TIME command) as well as the total running time since the job was initialized.

### Characteristics

The TIME command:

Leaves the terminal in monitor mode.

Does not require LOGIN when requesting time for another user's logged in job.

Associated Messages

Refer to Chapter 4.

Example

```
.TIME)
0.38
0.38
KILO-CORE-SEC=1
```

The command is given for the first time after LOGIN; therefore, the incremental time equals the total time since LOGIN.

```
.TI)
0.00
0.38
KILO-CORE-SEC=1
```

```
.DIR/F)
NEW505 BAK      DSKB: [10,770]
NEW505 RND
FOO   SFD
C     MAC      DSKC:
PETALS F4
PETALS SAV
CSHELL MAC
CIO   MAC
TRYCIO MAC
METER RND
SURFIT ALG
```

The DIRECT command took .40 seconds of runtime and 5 kilo-core-seconds.

```
.TI)
0.40
0.78
KILO-CORE-SEC=6
```

.

## TPUNCH Command

### Function

The TPUNCH command is used to place entries into the paper-tape punch output queue. This command is equivalent to the following form of the QUEUE command:

QUEUE PTP: jobname = list of input specifications.

The TPUNCH command can be further abbreviated to

PUNCH jobname = list of input specifications.

However, individual installations may redefine PUNCH to mean output to the card-punch queue instead of the paper-tape punch queue.

### Command Format

TPUNCH jobname = list of input specifications

jobname = name of the job being entered into the queue. The default is the name of the first file in the request not the name of the first file given. These differ when the first file given does not yet exist.

input specifications = a single file specification or a string of file specifications, separated by commas, for the disk files being processed. A file specification is in the form dev:file.ext [directory].

dev: = any disk file structure to which PTPSPL will have access; the default is DSK:.

file.ext = names of the files. The filename is optional. The default for the first filename is \*, the default for subsequent files is the last filename used. The extension can be omitted; the default is .PTP.

[directory] = a directory to which the user has access; the user's directory is assumed if none is specified.

The wildcard construction can be used for the input specifications.

The user can obtain the listing of entries in the paper-tape punch queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., TPUNCH [400,27]. If no arguments appear in the command string (i.e., only the command name is given), all entries in the paper-tape punch queue for all jobs are listed.

Switches that aid in constructing the queue entry can also appear as part of the input specifications. These switches are divided into three categories:

1. Queue-operation - Only one of these switches can be placed in the command string because they define the type of queue request. The switch used can appear anywhere in the command string.
2. General - Each switch in this category can appear only once in the command string because these switches affect the entire request. The switch used can appear anywhere in the command string.
3. File control - Any number of these switches can appear in the command string because they are specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the filename, it becomes the default for subsequent files. For example, the command string

TPUNCH FILEA, FILEB/DISP:REN, FILEC

indicates that the DISPOSE switch applies only to FILEB. The command string

TPUNCH /DISP:REN FILEA, FILEB, FILEC

indicates that the DISPOSE switch applies to all three files.

The following switches can be used with the TPUNCH command. Note if an argument to a switch is omitted, the colon preceding the argument must also be omitted. Otherwise the argument is assumed to be zero and not the default value.

SWITCH	CATEGORY	EXPLANATION
/AFTER:tt	General	Process the request after the specified time; tt is either in the form of hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting AFTER time must be less than the DEADLINE time. If the switch, or the value of the switch, is omitted, no AFTER constraints are assumed.
/BEFORE:t	General	Queue only the files with a creation date before time t where t is in the form dd-mmm-yy.
/BEGIN:n	File Control	Start the output on the nth foot of tape. The default is to begin output on the first foot.
/COPIES:n	File Control	Repeat the output the specified number of times. N must be less than 64. If more than 63 copies are needed, two separate requests must be made. If this switch is omitted, one copy is made.

SWITCH	CATEGORY	EXPLANATION
/CREATE	Queue Operation	Make a new entry into the paper-tape output queue. This switch is the default for the queue-operation switches.
/DEADLINE:tt	General	Process the request before the specified time; tt is either in the form hhmm (time of day) or +hhmm (time later than the current time). A colon may be used to separate the hours from the minutes (e.g., hh:mm). The resulting DEADLINE time must be greater than the AFTER time. If the switch, or the value of the switch, is omitted, no DEADLINE constraints are assumed.
/DISPOSE:DELETE	File Control	Delete the file after it has been punched.
/DISPOSE:PRESERVE	File Control	Do not delete the file after it has been punched. This is the default of all files except files with extensions of .LST, .TMP, and if the protection is 0xx, .PTP.
/DISPOSE:RENAME	File Control	Remove the file from the specified directory immediately. The DELETE option of the DISPOSE switch does not delete a file until it has been punched. Thus, the file continues to occupy space in the specified directory. The RENAME option frees the space immediately. If omitted, this is the default for files with extensions .LST, .TMP, and if the protection is 0xx, .PTP.
/F	Queue Operation	List the entries in the paper-tape punch queue, but do not update the queues. Therefore, the list may not be an up-to-date listing, but the listing will be faster than with /LIST.
/FILE:ASCII	File Control	Indicate that the file format is ASCII text. This is the default.
/FILE:ELEVEN	File Control	Indicate that the file format is MACX11 binary format.
/KILL	Queue Operation	Remove the specified entry from the paper-tape punch queue. This switch can be used for deleting a previously submitted request as long as the request has not been started by the paper-tape punch spooler.



SWITCH	CATEGORY	EXPLANATION
/LIMIT:n	General	Limit the output to the specified number of feet. The default is 10*B+20 feet, where B is the number of blocks in the request.
/LIST	Queue Operation	List the entries in the paper-tape punch queue; if the switch, along with all other switches, is omitted, all entries for all jobs of all users are listed.
/MODIFY	Queue Operation	Alter the specified parameters in the job. This switch requires that the user have access rights to the job. It can be used for altering a previously submitted request as long as the request has not been started by the spooler.
/NEW	File Control	Accept the request even if the file does not yet exist.
/NOTE:a	File Control	Punch the specified text (a) in the output.
/NULL	General	Do not output an error message if there are no files in the request and do not create a queue entry. This is assumed at KJOB time.
/OKNONE	File Control	Do not output message if no files match the wildcard construction. This is assumed at KJOB time. However, a totally null queue request produces a fatal error message.
/PHYSICAL	File Control	Suppress logical device name assignments for the device specified.
/PRIORITY:n	General	Assign the specified external priority (n = 0 to 62) to the request. The larger the number, the greater priority the job has. The default is 10 if no switch is given and 20 if the switch is specified without a value.
/PROTECT:nnn	General	Assign the protection nnn (octal) to the job. If the switch, or the value of the switch, is omitted, the standard protection is assumed.

SWITCH	CATEGORY	EXPLANATION
/REMOVE	File Control	Remove the file from the queue. This switch is valid only with the /MODIFY switch and can be used to remove a previously submitted file as long as the spooler has not started processing the request.
/SEQ:n	General	Specify a sequence number to help identify a request to be modified or deleted.
/SINCE:t	General	Queue only the files with creation dates after the specified time t where t is in the form dd-mmm-yy.
/START:n	File Control	Begin on the nth line of the file. If the switch, or the value of the switch, is omitted, the spooler starts with the first line.
/STRS	File Control	Search for the file on all file structures in the search list and take each occurrence. The default is to take just the first occurrence.
/TAPE:ASCII	File Control	Punch the tape in ASCII mode. If the /TAPE switch is not specified, the file is punched according to the data mode of the file.
/TAPE:BINARY	File Control	Punch the tape in binary mode. If the /TAPE switch is not specified, the file is punched according to the data mode of the file.
/TAPE:IBINARY	File Control	Punch the tape in image-binary mode. If the /TAPE switch is not specified, the file is punched according to the data mode of the file.
/TAPE:IMAGE	File Control	Punch the tape in image mode. If the /TAPE switch is not specified, the file is punched according to the data mode of the file.
/UNPRESERVED	General	Output the files only if they are not preserved (i.e., the first digit is 0). This switch avoids redundant punching.

## Characteristics

The TPUNCH command:

Leaves the terminal in monitor mode.

Runs the QUEUE program, thereby destroying the user's core image.

Does not require LOGIN when only queued listings are desired.

Depends on FTQCOM which is normally absent in the DECsystem-1040.

## Associated Messages

Refer to Chapter 4.

## Examples

```
.TPUNCH TENDMP.REL/TAPE:BINARY/COPIES:5)
```

Punch 5 copies, in binary mode, of the file DSK:TENDMP.REL.

## TYPE Command (1)

### Function

The TYPE command directs PIP to type the contents of the named source file(s) on the user's terminal. Note that if more than one file is requested in the command string, the files are typed out one after another with no indication of the beginning and ending of a file.

### Command Format

#### TYPE list

list = a single file specification or a string of file specifications separated by commas. The filename (including any extension) is required for a directory device.

In addition, the full wildcard construction can be used for the filename and/or the extension.

Switches can be passed to PIP by enclosing them in parentheses in the TYPE command string. When COMPIL interprets the command string, it passes the switches on to PIP.

### Characteristics

The TYPE command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core area.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Examples

```
.TYPE FILEA,DAT0:FILEB,MAC )  
.TYPE *.TMP,DTA4:C )
```

---

1

This command runs the COMPIL program, which interprets the command before running PIP.

## UNLOAD Command

### Function

The UNLOAD command rewinds and unloads a tape, either magnetic tape or DECTape. When unloading magnetic tape, WATCH statistics are printed on the operator's terminal. These statistics are also printed on the user's terminal by default.

### Command Format

UNLOAD dev:

dev: = a magnetic tape (MTAn) or a DECTape (DTAn).

### Characteristics

The UNLOAD command:

Leaves the terminal in monitor mode.

Runs the COMPIL program, thereby destroying the user's core image.

### Associated Messages

Refer to Chapter 4.

### Examples

```
.UNLOAD DTA7:)
```

```
.UNL MTA3:)
```

## USESTAT Command

### Function

The USESTAT command returns status information pertinent to the current user. This status information is printed on one line in eight columns. The information given represents:

- incremental day time in seconds
- incremental run time in seconds
- incremental disk reads
- incremental disk writes
- program name
- core size
- job state
- program counter (PC)

Refer to Paragraph 1.3.1.1 for a description of the status information returned as a result of the USESTAT command (or ^T).

The results obtained from issuing the USESTAT command may also be obtained by issuing a CONTROL-T (^T).

### Command Format

USESTAT

### Characteristics

The USESTAT command:

- Leaves the terminal in monitor mode.

## Function

The VERSION command prints the version number of the program in the user's core area (i.e., the last program run implicitly or explicitly). One use of this command is to determine the program that output a message to the user. If the terminal is still in user mode (i.e., a character other than a period was output) after the message, the user can type the following

```
^C          (two ^C's if the program is not waiting
.VERSION    for input)
```

The monitor returns with the name of the program in core (i.e. the one presumed to have output the message) and the version number of that program. After receiving the information, the user can type .CONT to return the terminal to user mode. If the message was a fatal message (i.e., a period was output after the message). ^C need not be typed since the terminal is already in monitor mode. In most cases, the user cannot type .CONT after a fatal error message.

The version number is obtained from .JBVER and .JBHVR in the job data area and is printed in standard format. Similar output is automatically generated by the SET WATCH VERSION command (refer to the SET WATCH command description). The output from these two commands is in one of the following representations:

low + high	The low and high segments are different.
low	There is only a low segment.
low +	The low and high segments are the same.
+(1)	A GETSEG UVO has been done to a high segment which matches the low segment.
+ high(1)	A GETSEG UVO has been done to a high segment which does not match the low segment.
blank(1)	The high segment has been released.

With the VERSION command, the low and high segments are represented in the format

```
name version
```

With the SET WATCH VERSION command, the low and high segments are represented in one of three formats:

---

1

Output only from the SET WATCH VERSION command.

name version	The program is not from SYS:
:name version	The output is the result of a SETNAM UOO (e.g., at the end of loading).
S:name version	The program is a program loaded from the system device (actual SYS: not logical device SYS:).

The name is a SIXBIT name and the version is in standard format. When printing the version number, the standard format is:

major version minor version (edit) - group who modified program last

The major version is octal; the minor version is alphabetic; the edit is octal and enclosed in parentheses; and the group who last modified the program is octal and preceded by a hyphen (0 = DEC development, 1 = all other DEC personnel, and 2-7 = customer use). There are no spaces separating the items, and if an item is zero, it does not appear in print. The parentheses and hyphen also do not appear in print if the corresponding item is zero. The following are examples of version numbers output in standard format.

10B(335)-1	major version 10, minor version B, edit number 335, group that modified program last 1.
7(5)	major version 7, minor version 0, edit number 5, group that modified program last 0.
54A	major version 54, minor version A, edit number 0, group that modified program last 0.

When running under a virtual memory system, DAEMON will be called on execution of the VERSION command if the first page of the high segment is paged out.

#### Command Format

VERSION

#### Characteristics

The VERSION command:

Leaves the terminal in monitor mode.



Examples

.R TECO )

\*^C

.VERSION )  
TECO 22 (64) +

.TYPE SAMPL.TXT )  
THIS IS A TEXT FILE

.VERSION  
PIP 31 (35) +

## WHERE Command

### Function

The WHERE command enables the user to determine the station at which a specific peripheral device is located. If the station of a particular terminal is requested, the number returned is the physical location of the terminal which may or may not be the location of the controlling job. This depends on whether the user changed his job's logical location with the LOCATE command.

### Command Format

WHERE devn

dev = any physical device name and n is the unit number.

### Characteristics

The WHERE command:

Leaves the terminal in monitor mode.

Does not require LOGIN.

Depends on FTREM which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Examples

`.WHERE CDR2:)`  
1 :station of CDR2.

`.WH TTY:)`  
2 :station job's terminal.  
:(physical terminal location).

`.WHE OPR:)`  
4 :station of job issuing command.  
:(logical terminal location).

`.WHERE CTY:)`  
I :central station.

## ZERO Command (1)

### Function

The ZERO command clears the directory of the output device. This command is equivalent to the following PIP command string:

```
dev: /Z_
```

### Command Format

ZERO dev:

dev: = a DECtape (DTAn) or a disk (DSK). This argument is required.

A directory name can be specified with ZERO DSK: and if the user has access to the specified directory, the directory is zeroed. If no directory is specified, the user's directory is assumed.

The only acceptable abbreviation of this command is ZER.

### Characteristics

The ZERO command:

Leaves the terminal in monitor mode.

Runs the PIP program, thereby destroying the user's core image.

Depends on FTCCLX which is normally absent in the DECsystem-1040.

### Associated Messages

Refer to Chapter 4.

### Examples

```
_ZER DTA4: )  
_ZERO DSK: )  
_ZER DSK: [27,401)
```

---

1

This command runs the COMPIL program, which interprets the command before running the PIP program.



## CHAPTER 3

### Batch System Commands

The Batch System, operating under the control of the DECsystem-10 Operating System, increases system throughput by processing jobs that do not require human interaction. Types of jobs best suited for a batch environment are: large and long-running jobs, jobs that require large amounts of data, frequently run production jobs, and jobs that require little or no interaction with the user. Up to 14 Batch jobs can be processed concurrently without adversely affecting the running of timesharing jobs. Batch jobs may be entered from

1. Local devices
2. Remote devices
3. Interactive terminals.

#### 3.1 BATCH COMPONENTS

The Batch System consists of a group of programs; some are used for Batch operations only, others are available for various operations of the total computing system.

The individual Batch components are: the input spooler, SPRINT-10 ; the Queue Manager, QMANGR; the Batch Controller, BATCON; and the output spoolers, LPTSPL (line printer), CDPSPL (card punch), PLTSPL (plotter), and PTPSPL (paper-tape punch).

##### 3.1.1 SPRINT-10

SPRINT-10 is responsible for

1. reading a sequential input stream from an input device,
2. separating the input by placing it in files according to the control cards contained in the input stream,
3. creating the job's log file and entering a report of its processing, and
4. entering the job into the Batch input queue.

When input is from the card reader, SPRINT-10 accepts ASCII, binary, and 026 Hollerith code. (Refer to Appendix B for tables of card codes.) The input is read in image mode, and SPRINT-10 converts it to one of the mentioned codes. If input is from any other device, only ASCII code is accepted.

SPRINT-10 creates three types of files during its copying of the input data: the user's data files, the Batch control file, and the job's log file. The data jobs are created according to the control cards in the input and are placed into the user's disk area. Programs and data are

copied into these files and are passed to the job, while it is running, by the Batch Controller. Refer to Paragraph 3.3 for the description of the control cards that cause SPRINT-10 to copy information into these files.

A user control file is created for each valid job and is subsequently processed by the Batch Controller. This file contains all monitor level and user level commands encountered in the input. SPRINT-10 enters commands resulting from the processing of certain control cards and any information that does not follow specific control card format. The control file is placed in the user's disk area. Refer to Paragraph 3.4 for a description of the Batch Controller commands that can be entered into the control file.

The job's log file contains a report SPRINT-10's processing, along with a record of any operator intervention during its operation. This file is in the user's disk area along with the other SPRINT-10-created files and is deleted after it is printed by the line printer spooler.

### **3.1.2 The Queue Manager**

The Queue Manager, QMANGR, is the program that schedules jobs and maintains system queues. When SPRINT-10 finishes processing a job, it makes an entry into the Batch input queue. The Queue Manager computes and dynamically revises priorities for the job and notifies the Batch Controller when the job is to be run. Jobs are scheduled for running according to the parameters pertaining to each job and the priorities established by the system. While the job is running, its queue entry is flagged to show it is in use, but the entry is not deleted from the queue until the job terminates. When the job is logged off the system, an output queue entry is usually made and the entry in the input queue is deleted. The Queue Manager again schedules the job's output and deletes the job's output queue entry only when the output is completely finished.

### **3.1.3 The Batch Controller**

The Batch Controller, BATCON, controls all jobs entered into the Batch System. It reads the control file created by SPRINT-10 or the user and initiates and controls the running of the job by passing data and system program commands directly to it.

Monitor commands are examined by the Batch Controller and passed to the monitor for action. The Controller determines the destination of commands by interpreting the character in column 1 in each line of the control file. If column 1 contains a space or a tab, the spaces are ignored until a nonspace character is encountered. If column 1 contains an alphabetic or numeric character, the line is either at monitor command level or at user command level. If column 1 contains a special character, the Batch Controller interprets the line as follows:

\$ (dollar sign) - The interpretation depends upon the character in column 2.

If column 2 contains an alphabetic character, the line is copied to the log file as a comment because it is a SPRINT-10 control line and has already been processed.

If column 2 contains a numeric or special character, the line is treated as data.

If column 2 contains a dollar sign (\$), the initial dollar sign is suppressed and the line is treated as data.

If column 2 contains a line feed, vertical tab, or form feed, a blank line is entered into the log file.

. (period) - The interpretation depends upon the character in column 2.

If column 2 contains an alphabetic character, the line is treated as a monitor command and the period is suppressed.

If column 2 contains a nonalphabetic character, the line is treated as data with the period as part of the data.

\* (asterisk) - The line is treated as a user-level command or program data and the asterisk is suppressed. This is the standard input data method for most system programs.

= (equal sign) - The line is treated as a user level command or program data. The equal sign is suppressed and final spaces and the end of the line are suppressed (i.e., not passed to the program). This line normally indicates a DDT or TECO command because these commands terminate with special characters rather than the end of the line and would not function properly if the end of the line were passed.

; (semicolon) - The line is treated as a comment to the log file.

% (percent sign) - The line is treated as part of a command level statement label. The percent sign is normally reserved for DEC use. If % is encountered when the job has had no error, the control file is advanced, unless a %FIN is encountered. In this case, the %FIN is executed. Refer to the discussion of the .IF command in Paragraph 3.4.5.

The Batch Controller does not examine the contents of any lines in the control file other than those destined for the monitor. However, when it encounters an up-arrow (↑), it converts the up-arrow as follows:

If the character following the up-arrow is a numeric character, the up-arrow and the digit are passed to the job.

If the character following the up-arrow is an alphabetic character, the up-arrow and the character are translated to a control character; e.g., ↑A is translated to CTRL-A.

If the character following the up-arrow is another up-arrow, the first up-arrow is ignored and the second up-arrow is treated as an up-arrow; e.g., ↑↑A is treated as ↑A (up-arrow A) and ↑↑↑A is treated as ↑↑A (up-arrow up-arrow A).

If the job is requesting input and is in monitor mode, the control file is read until a command or intermediate level is found. If a job is requesting input at the data mode level (refer to Paragraph 1.2 for a detailed explanation of the various modes) and the next line is a monitor command, the Batch Controller inserts a control-C.

It is possible to enter a conversational mode between a running program and the system operator through BATCON. In this mode, lines of output are copied to the terminal controlling BATCON (normally the system operator's console) and requests for input are

honed by the same terminal rather than by the control file. The level at which the input is requested determines whether the input interpreted by the operator is interpreted at the monitor mode, user mode, or data mode level. The operator may not enter BATCON commands such as GOTO. There are two commands available to the user to enter and leave dialogue mode. The .OPERATOR command (Paragraph 3.4.8) is used to enter dialogue mode and .NOOPERATOR (Paragraph 3.4.7) is the command to leave dialogue mode.

A Batch user may not issue the following monitor commands when his job is operating in batch mode: ATTACH, DETACH, CCONT, CSTART, SET TTY, TTY, SET TIME, and SEND. If these commands are used, the line is suppressed and flagged with the label BAERR in the log file and the job is continued. All other monitor commands and system program commands may be used by a job operating in batch mode.

The Batch Controller makes entries to the log file to record its processing of the control file and the job.

### 3.1.4 The Output Spoolers

The output spoolers receive job output that has been placed into the output queues by the Queue Manager. Usually a job's output is placed in a line printer queue to be printed at a later time by the LPTSPL spooling program at the same station from which the input was received. The output filenames are in the form QxxSnn.LPT, where xx is a random number, and nn is the station number of the printer where the job is currently located. However, the user can also specify other output devices either in his programs within his job or by means of the QUEUE monitor command in his job. The first method causes output to the card punch, paper-tape punch, or plotter to be automatically spooled by the system. The second specifies nonstandard output spooling to any of the spooling devices.

## 3.2 SUBMITTING JOBS

A job is a unit that consists of one step or a group of steps. It can contain:

1. a single program and its related data, or several programs and their data, and
2. the monitor and user-level commands that are required to control the programs.

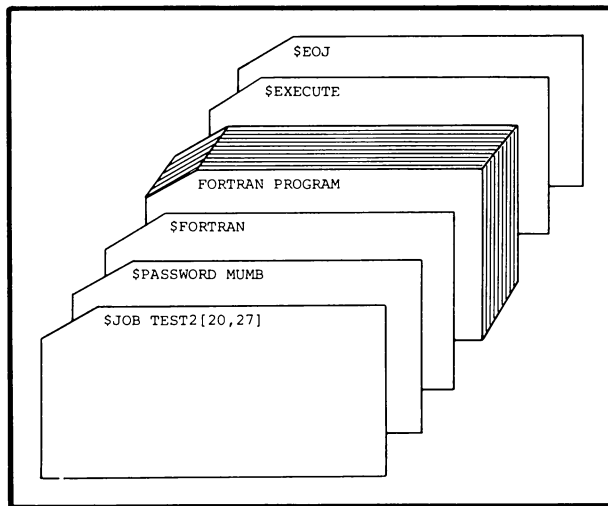
The Batch system allows the user to submit his job by one of the following three methods:

1. The user punches his job on cards, inserts control cards to SPRINT-10, and leaves his cards at the designated place for the operator to run (refer to Paragraph 3.2.1).
2. The user creates his job as a file for input to SPRINT-10 (instead of having his job on cards) and then runs SPRINT-10 himself (refer to Paragraph 3.2.2).
3. The user bypasses SPRINT-10 by creating his own control file on disk for the Batch Controller and then enters his job into the Batch input queue from his terminal (refer to Paragraph 3.2.3).



### 3.2.1 Submitting a Job with Cards

With this method, a job is submitted via a deck of cards, bounded by the control cards that mark its beginning and end. Other control cards to SPRINT-10 are interspersed among the card deck to direct SPRINT-10's processing. Figure 3-1 shows a job containing the appropriate control cards to SPRINT-10. This job compiles, loads, executes, and lists a FORTRAN program.



This figure illustrates, and Paragraphs 3.2.1.3 and 3.2.2 describe, a job which happens to contain a FORTRAN program. With the substitution of the appropriate \$-language card for the \$FORTRAN card, this arrangement would be equally valid for other supported languages.

Figure 3-1 Typical Job on Cards

**3.2.1.1 The \$JOB Card** – This card notifies SPRINT-10 that a job is to be processed. SPRINT-10 creates a control file into which commands are placed for the Batch Controller and a log file on the disk. The first argument (TEST2) shown on this card is the user-assigned name for the job; the second argument ([20,27]) is the project-programmer number of the user. For a description of switches which can be used on this card, refer to Paragraph 3.3.13.

**3.2.1.2 The \$PASSWORD Card** – This card contains the PASSWORD associated with the project-programmer number specified on the \$JOB card. In Figure 3-1, the PASSWORD is MUMB, which was assigned to the user by the system manager. Refer to Paragraph 3.3.16 for more information on the \$PASSWORD card.

**3.2.1.3 The \$FORTRAN Card** - This card causes SPRINT-10 to insert a COMPILE monitor command (refer to Chapter 2) into the control file in order to cause the program to be compiled. Immediately following the \$FORTRAN card is the FORTRAN source program to be compiled. The source program is read into a disk file with the specified filename (or a default name if a filename is not given) and with an extension of .FOR (or .F4). Refer to Paragraph 3.3.11 for more information on the \$FORTRAN card.

**3.2.1.4 The \$EXECUTE Card** – The card after the FORTRAN program is the \$EXECUTE card. This card causes SPRINT-10 to insert an EXECUTE monitor command (refer to Chapter 2) into the control file in order to load and then execute the previously compiled program. Refer to Paragraph 3.3.10 for additional information on this card.

**3.2.1.5 The \$EOJ Card** – The last card shown in the example is the \$EOJ card. It signals the end of the job. The standard end-of-file card (punches in rows 12, 11, 0, 1, 6, 7, 8, 9 of columns 1 and 80), can also be used to terminate the job. However, it is recommended that the \$EOJ card be used when running a job from a remote batch station. Refer to Paragraph 3.3 for more information about these cards.

**3.2.1.6 Output** – Once the program is punched on cards, the card deck is submitted to the operator, who in turn stacks the job in the card reader. The user receives his output in the form of line printer listings. Refer to Paragraph 3.5 for an explanation of the job output.

The SPRINT-10 control cards shown in Figure 3-1 are just a few of the control cards available to the user. For a complete description of all the SPRINT-10 control cards, refer to Paragraph 3.3.

### 3.2.2 Submitting a Job with a File

With this method, a job is submitted via a file contained on any input device that supports ASCII code. This file contains the program and data with card images of the control cards for SPRINT-10. The following example shows the creation of a disk file containing a FORTRAN program and card images of SPRINT-10 commands. Note that it corresponds to the card example in Paragraph 3.2.1.

```
.LOGIN 20,27
JOB17 5S04 TTY11
PASSWORD:
1020 15-MAR-72 WED
.MAKE JOBFIL
*I $JOB TEST2, [20,27]
$FORTRAN
C FORTRAN PROGRAM GOES HERE
$EXECUTE
$$
*EX$$
```

**3.2.2.1 Image of the \$JOB Card** – The first line of the file is an image of the \$JOB card. Note that the \$ character must be the first character of the line in order for SPRINT-10 to recognize it as a control command. This line causes a control file and a log file to be created on the disk when SPRINT-10 is run. The first argument (TEST2) is the user-assigned name for the job; the second ([20,27]) is the project-programmer number of the user. The

\$PASSWORD card image is not needed because the user is already logged-in when creating the input file. For additional information on the \$JOB card, refer the Paragrah 3.3.13.

**3.2.2.2 Image of the \$FORTRAN Card** - This line causes SPRINT-10 to insert a COMPILE monitor command (refer to Chapter 2) into the control file in order to compile the program. The source program follows immediately and is read into a disk file with the specified filename (or a default name if a filename is not given) and with an extension of .FOR (or .F4). Refer to Paragraph 3.3.11 for more information on the \$FORTRAN card.

**3.2.2.3 Image of the \$EXECUTE Card** - This line causes SPRINT-10 to insert an EXECUTE monitor command (refer to Chapter 2) into the control file in order to load and execute the program.

**3.2.2.4 Running SPRINT-10** - Once the file is created and SPRINT-10 is run by the user, it processes the user-created file in the same manner as it processes input files of jobs entered directly by the operator. The user runs SPRINT-10 by typing

```
.R SPRINT
```

SPRINT-10 responds with a slash, and then the user types in the following command

```
/START dev:file.ext
```

where dev: is the name of the device containing the input file for SPRINT-10 and file.ext is the name of the file. If the device, filename and extension are omitted, the card-reader is assumed. (That is, the command, /START is interpreted as /START CDR:.) If the filename is typed and the device is omitted, disk is assumed. (That is, the command /START file.ext, is interpreted as /START DSK:file.ext.) Using the above file, the command is

```
/START JOBFIL
```

and SPRINT-10 responds with

```
!
```

When SPRINT-10 has completed its processing (i.e., when it has created the control and log files and has entered the job into the Batch input queue), it responds with

```
[SPRINT IS RESET]
```

```
/
```

indicating its readiness to accept another file. At this point, the user can enter another file or return to monitor mode with a  $\uparrow$ C.

The card images shown in the preceding example are only a few of the SPRINT-10 control card images available. Refer to Paragraph 3.3 for a complete description of all the control cards.

### 3.2.3 Submitting a Job with a Control File to the Batch Controller

With this method, a job is submitted via the steps within a control file to the Batch Controller. The file must be a disk file and is created with a system editor. Since this file is processed directly by the Batch Controller, control card images are not used. The control file consists of monitor commands, user program commands, comments, and sequence control statements. Refer to Paragraph 3.4 for a description of control file commands. The following is an example of creating a control file. It assumes that a file named DATA.F4 already exists on disk.

```
.MAKE JOB.CTL

*I.EXECUTE /COMPILE DATA.F4 /LIST

$$
EX$$
```

Once the control file is created, the user can enter the job into the Batch input queue one of three ways:

1. SUBMIT jobname = control file, log file  
refer to the SUBMIT command in Chapter 2.
2. QUEUE INP:jobname = control file, log file  
refer to the QUEUE command in Chapter 2.

3. R QUEUE

refer to the QUEUE specification in Notebook 8 of the DECsystem-10 Software Notebooks.

### 3.2.4 Interjob Dependency

Jobs are not necessarily run in the order that they are read into the Batch System. Priorities stipulated by the user on the \$JOB card (refer to Paragraph 3.3.13) and additional parameters set by the Batch System are dynamically computed by the Queue Manager to determine in what order the jobs are run. However, it is often useful to submit several jobs that must be run in a specific order, for example, one job updates a master file and another job processes it. Therefore, the running of one job is dependent upon the running of the other. Although these jobs could be combined into one large job, it is sometimes necessary to keep them distinct; i.e., they might be submitted by different people at different times. Because the jobs in the Batch System are run in order of priority, the user specifies an additional priority, an initial dependency count, on the \$JOB card of the dependent job. This dependency count becomes part of the queue entry. Any input queue entry that has a dependency count greater than zero cannot be scheduled. When the count becomes zero, the job is scheduled, based upon the time it was submitted and the time that the dependency count became zero. If the dependency count becomes negative, an advisory message is sent to the issuing job and to the dependent job. The dependency count can be altered by including the QUEUE command as part of any job upon which the dependent job is waiting. (Refer to the QUEUE monitor command.) The QUEUE command switch that allows the user to change the dependency

count of another job is the /MODIFY/DEPEND:nn switch. If the user specifies a plus or minus sign before the count (nn), that number is subtracted from or added to the dependent job's count. If the user does not specify a sign, the dependent job's count is changed to the count specified in the /MODIFY/DEPEND:switch.

### 3.3 SPRINT-10 CONTROL CARDS

Control cards are interspersed among the input stream to aid SPRINT-10 in separating the input into the appropriate files, either the user's data files or the control file processed by the Batch Controller. The control cards contain a dollar sign (\$) in column 1 and an alphabetic character in column 2. These are the only cards read and interpreted by SPRINT-10; the remainder of the input is separated and placed into the appropriate file. Of these control cards, the \$DATA, DECK, \$-language and \$RELOCATABLE cards are also referred to collectively as File Cards, in that they all cause subsequent cards up to the next SPRINT-10 control card (except \$MODE) to be placed into a disk file. Note that if the user creates his own control file, he bypasses SPRINT-10 and, therefore, does not use these control cards.

Only the first part of the command name or switch need be specified; as long as the name is unique within its class, it is accepted. The first three characters of a command name are generally sufficient to ensure uniqueness. The standard comment and continuation conventions for the system can be used on the control card.

A comment is preceded by a semicolon; and all the characters following the semicolon through the end of card are treated as comments. While a comment field cannot actually be continued, the same effect can be accomplished by the use of an additional card which contains only a comment.

```
$DECK MYFILE.ONE ;THIS IS A COMMENT
```

```
;$THIS IS A COMMENT
```

Operative information may be continued by placing a hyphen as the last non-TAB, non-space, or non-comment character in the operative field. That is, the hyphen may be followed by one of the following:

1. the end of the card, or
2. a string of spaces and/or TABs followed by the end of card, or
3. a semicolon followed by a comment.

The following two examples illustrate the continuation of operative information; first without comments and then with comments, respectively.

1. \$DECK MYFILE.ONE /ASCII -  
/SUPPRESS
2. \$DECK MYFILE.ONE - ;THIS IS A COMMENT  
/ASCII/SUPPRESS

All defaults for control card parameters are installation parameters.

The end-of-file is used to signal the end of the job. When input is from the card reader, a \$EOJ or end-of-file card is used. Column 1 of the end-of-file card contains punches in rows 12, 11, 0, 6, 7, 8 and 9 with rows 2, 3, 4 and 5 blank. The recommended form of this card also has 80 columns punched as designated above, so that the card can be recognized in any orientation. However, it is recommended that the \$EOJ card be used when running a job from a remote batch station.

When devices other than the card reader are used for input, the standard end-of-file for each device is treated by SPRINT-10 as the end of the job.

### 3.3.1 \$ALGOL

#### Function

This card precedes an ALGOL source program and causes SPRINT-10 to copy the program onto disk and to insert a COMPILE monitor command into the control file. The source program is ended by any SPRINT-10 control card other than a \$MODE.

When the job is run, the specified program is compiled and temporary relocatable binary and listing files are created. The source, binary and listing files can be made permanent if the user renames them to change their protection. (Refer to the RENAME command in Chapter 2.) The listing file is printed as part of the job's output.

The \$ALGOL card does not cause execution of the program. Execution is initiated by an EXECUTE monitor command or by a \$DATA or a \$EXECUTE card. (Refer to the EXECUTE command in Chapter 2 or paragraphs 3.3.4 and 3.3.10 in this chapter, respectively.)

Processor switches can be passed to the ALGOL compiler by including them in the command string. Refer to Paragraph 1.5.7 for a description of the ALGOL processor switches.

#### Card Format

\$ALGOL dev:name.ext [proj,prog] (processor switches)/S(1)/S(2).../S(n)

dev: = a file structure name. If omitted, DSK is assumed.

name.ext = the name of the file to be created on disk. If omitted, SPRINT-10 assigns the filename LN???? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique filename) with the extension .ALG. However, when the user specifies only a filename (with no extension), SPRINT-10 will not append any extension. The source and relocatable binary files will be explicitly deleted at the end of the job if no filename was specified. Where the user has specified a filename, the disposition of the files depends on the protection the user has given them.

- [proj,prog] = a directory name other than that specified on the \$JOB card. If omitted, the project-programmer number on the \$JOB card is used.
- (processor switches) = the switches to be passed to the ALGOL compiler. They must be enclosed in parentheses and the slash cannot appear in connection with these switches.
- /S(1)/S(2)...S(n) = the switches that control the mode of input interpretation and the listing of the compiled program.

Switch	Meaning
/ASCII	The input is read in ASCII mode. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.
/NOLIST	No listing file of the program is created. If this switch is not specified, a listing file will be created.
/NOSUPPRESS	Trailing blanks are not suppressed. The default is no suppression.
/SUPPRESS	Trailing blanks are suppressed. The default is no suppression.
/WIDTH:nn	Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks. If this switch is not specified or if no argument is specified, then SPRINT-10 will assume a value of 80.

Switch	Meaning
--------	---------

/026	
------	--

The card deck is read in 026 card code. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13. This switch applies only to card-reader input. Input from other devices must be read in ASCII code; otherwise, an error message is written in the log file and the job is terminated.

### 3.3.2 \$BLISS

#### Function

This card precedes a BLISS source program and causes SPRINT-10 to copy the program onto disk and to insert a COMPILE monitor command into the control file. The source program is ended by any SPRINT-10 control card other than a \$MODE.

When the job is run, the specified program is compiled and temporary relocatable binary and listing files are created. The source, binary and listing files can be made permanent if the user renames them to change their protection. (Refer to the RENAME command in Chapter 2.) The listing file is printed as part of the job's output.

The \$BLISS card does not cause execution of the program. Execution is initiated by an EXECUTE monitor command or by a \$DATA or a \$EXECUTE card. (Refer to the EXECUTE command in Chapter 2 or paragraphs 3.3.4 and 3.3.10 in this chapter, respectively.)

Processor switches can be passed to the BLISS compiler by including them in the command string.

Refer to Chapter 3 of the BLISS-10 Reference Manual and to the COMPILE command for a description of the BLISS processor switches.

#### Card Format

\$BLISS dev:name.ext [proj.prog] (processor switches)/S(1)/S(2).../S(n)

dev: = a file structure name. If omitted, DSK is assumed.



- name.ext = the name of the file to be created on disk. If omitted, SPRINT-10 assigns the filename LN???? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique filename) with the extension .BLI. However, when the user specifies only a filename (with no extension), SPRINT-10 will not append any extension. The source and relocatable binary files will be explicitly deleted at the end of the job if no filename was specified. Where the user has specified a filename, the disposition of the files depends on the protection the user has given them.
- [proj,prog] = a directory name other than that specified on the \$JOB card. If omitted, the project-programmer number on the \$JOB card is used.
- (processor switches) = the switches to be passed to the BLISS compiler. They must be enclosed in parentheses and the slash must not appear in connection with these switches.
- /S(1)/S(2).../S(n) = the switches that control the mode of input interpretation and the listing of the compiled program.

Switch	Meaning
/ASCII	The input is read in ASCII mode. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.
/NOLIST	No listing file of the program is created. If this switch is not specified, a listing file will be created.
/NOSUPPRESS	Trailing blanks are not suppressed. The default is no suppression.
/SUPPRESS	Trailing blanks are suppressed. The default is no suppression.

Switch	Meaning
/WIDTH:nn	Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks. If this switch is not specified or if no argument is specified, then SPRINT-10 will assume a value of 80.
/026	The card deck is read in 026 card code. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13. This switch applies only to card-reader input. Input from other devices must be read in ASCII code; otherwise, an error message is written in the log file and the job is terminated.

### 3.3.3 \$COBOL

#### Function

This card precedes a COBOL source program and causes SPRINT-10 to copy the program onto disk and to insert a COMPILE monitor command into the control file. The source program is ended by any SPRINT-10 control card other than a \$MODE.

When the job is run, the specified program is compiled and temporary relocatable binary and listing files are created. The source, binary and listing files can be made permanent if the user renames them to change their protection. (Refer to the RENAME command in Chapter 2.) The listing file is printed as part of the job's output.

The \$COBOL card does not cause execution of the program. Execution is initiated by an EXECUTE monitor command or by a \$DATA or a \$EXECUTE card. (Refer to the EXECUTE command in Chapter 2 or paragraphs 3.3.4 and 3.3.10 in this chapter, respectively.)

Processor switches can be passed to the COBOL compiler by including them in the command string. Refer to Paragraph 1.5.7 for a description of the COBOL processor switches.

## Card Format

\$COBOL dev:name.ext [proj,prog] (processor switches) /S(1)/S(2).../S(n)

dev:	= a file structure name. If omitted, DSK is assumed.
name.ext	= the name of the file to be created on disk. If omitted, SPRINT-10 assigns the filename LN???? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique filename) with the extension. CBL. However, when the user specifies only a filename (with no extension), SPRINT-10 will not append any extension. The source and relocatable binary files will be explicitly deleted at the end of the job if no filename was specified. Where the user has specified a filename, the disposition of the files depends on the protection the user has given them.
[proj,prog]	= a directory name other than that specified on the \$JOB card. If omitted, the project-programmer number on the \$JOB card is used.
(processor switches)	= the switches to be passed to the COBOL compiler. They must be enclosed in parentheses and the slash must not appear in connection with these switches.
/S(1)/S(2).../S(n)	= the switches that control the mode of input interpretation and the listing of the compiled program.

Switch	Meaning
--------	---------

/ASCII	
--------	--

	The inpt is read in ASCII mode. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.
--	--

/NOLIST	
---------	--

	No listing file of the program is created. If this switch is not specified, a listing file will be created.
--	---

Switch	Meaning
/NOSUPPRESS	Trailing blanks are not suppressed. The default is no suppression.
/SUPPRESS	Trailing blanks are suppressed. The default is no suppression.
/WIDTH:nn	Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks. If this switch is not specified or if no argument is specified, then SPRINT-10 will assume a value of 80.
/026	The card deck is read in 026 card code. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13. This switch applies only to card-reader input. Input from other devices must be read in ASCII code; otherwise, an error message is written in the log file and the job is terminated.

### 3.3.4 \$DATA

#### Function

This card is a File Card which precedes a data deck and causes SPRINT-10 to copy the data into a file on the user's disk area and to insert an EXECUTE monitor command into the control file.

SPRINT-10 maintains a list of filenames of all source or relocatable programs that have been processed since the beginning of the job or the last \$DATA or \$EXECUTE card read. Each time a program is copied by SPRINT-10, its name is placed in the list and given an extension

of .REL. When the \$DATA card is read, SPRINT-10 places an EXECUTE command into the control file and copies the filenames of the programs into the EXECUTE command string. On the next \$language, \$INCLUDE or \$RELOCATABLE card, SPRINT-10 clears the list of filenames so that the next entries into the list reflect only those filenames copied since the last \$DATA or \$EXECUTE card was read. When the job is run, the programs are loaded and executed. No compilation is performed because the programs are either in relocatable binary form or have been previously compiled because of the \$language card. If two \$DATA cards appear in a row, the same programs are reloaded and executed again.

### Card Format

\$DATA dev:name.ext [proj,prog] /S(1)/S(2).../S(n)

- dev: = a file structure name. If omitted, DSK is assumed.
- name.ext = the name of the file to be created. If omitted, SPRINT-10 creates the filename ??? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a \unique filename) with the extension .CDR and creates a spooled card-reader file. That is, the user may read the data file simply by referencing the card-reader from his program. However, when a filename is specified, the user must specifically read the data file from disk.
- [proj,prog] = the directory name if different from the one specified on the \$JOB card. If omitted, the project-programmer number specified on the \$JOB card is used.
- /S(1)/S(2).../S(n) = switches that control the mode of reading and interpreting of the input media.

Switch	Meaning
--------	---------

/ASCII

The input is read in ASCII mode. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.

/BINARY

The card deck is read in binary card form. This switch is ordinarily not necessary because the first column of each card is checked

Switch	Meaning
	for punches in rows 7 and 9. If these rows are punched, the card is read in binary. If no other card format switch is specified then the default is the mode indicated on the /HOLLERITH switch on the SJOB card. Refer to Paragraph 3.3.13.
/IMAGE:n	The card deck is read in image mode. The switch must be followed by a decimal number in the range 2 through 80. This causes ensuing cards to be read in image mode until either end-of-file is reached or a card is read that contains punches in all rows of column 1 and in all rows of column n. The SPRINT-10 control commands are not recognized when cards are read in image mode. If no value is given for n, then SPRINT-10 will assume 2. If no other card format switch is specified then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.
/MAP	This switch causes a loader map to be generated and printed. The default is no map.
/NOSUPPRESS	Trailing blanks are not suppressed. The default is no suppression.
/SUPPRESS	Trailing blanks are suppressed. The default is no suppression.

Switch	Meaning
/WIDTH:nn	Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks. If this switch is not specified or if no argument is specified, then SPRINT-10 will assume a value of 80.
/026	<p>The card deck is read in 026 card code. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13. This switch applies only to card-reader input. Input from other devices must be read in ASCII code; otherwise, an error message is written in the log file and the job is terminated.</p> <p>The modes ASCII, 026, and IMAGE are mutually exclusive modes for interpreting Hollerith punches. When one of these modes is set, it remains in effect until changed (refer to the \$MODE card) or the end of file is reached.</p>

The defaults for all modes are reset by the next \$MODE card or by individual switches in other control cards such as in the \$DECK card and \$JOB card.

## Restrictions

This card can be used only when the programs in the job have been entered with a \$language, \$INCLUDE, or \$RELOCATABLE card, since SPRINT-10 maintains a list of the filenames of programs that are input with these commands. If the user wishes only to have the programs compiled, no \$DATA or \$EXECUTE card or EXECUTE command should appear in the job.

### 3.3.5 \$DECK

#### Function

This card causes SPRINT-10 to copy all cards up to the next control card (except a \$MODE card) into a data file. The \$DECK card also has the capability of placing the specified file into various output queues.

#### Card Format

\$DECK dev:name.ext [proj,prog] /S(1)/S(2).../S(n)

dev: = a file structure name. The default is normally DSK.

name.ext = the user-assigned name and extension of the file to be created. If omitted, the filename DK??? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique filename) is created by SPRINT-10.

[proj,prog] = a disk area other than the one specified on the \$JOB card. If omitted, the project-programmer number specified on the \$JOB card is used.

/S(1)/S(2).../S(n) = switches that control the mode of reading and interpreting of the input media and the placement of the output (if desired) in the appropriate output queue.

Switch

Meaning

/ASCII

The input is read in ASCII mode. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.



Switch	Meaning
/BINARY	The card deck is read in binary card form. This switch is ordinarily not necessary because the first column of each card is checked for punches in rows 7 and 9. If these rows are punched, the card is read in binary. If no other card format switch is specified then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.
/CPUNCH	SPRINT-10 places the specified file into the card punch output queue. The default is that the file is not placed into the output queue.
/IMAGE:n	The card deck is read in image mode. The switch must be followed by a decimal number in the range 2 through 80. This causes ensuing cards to be read in image mode until either end-of-file is reached or a card is read that contains punches in all rows of column 1 and in all rows of column n. The SPRINT-10 control commands are not recognized when cards are read in image mode. If no value is given for n, then SPRINT-10 will assume 2. If no other card format switch is specified then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.

Switch	Meaning
/NOSUPPRESS	Trailing blanks are not suppressed. The default is no suppression.
/PLOT	SPRINT-10 places the specified file into the plotter output queue. The default is that the file is not placed in the output queue.
/PRINT	SPRINT-10 places the specified file into the line printer output queue. The default is that the file is not placed into the output queue.
/PUNCH	SPRINT-10 places the specified file into the output queue of the installation's default punch device. This is normally the paper-tape punch. If this switch is not specified, the file is not placed in the output queue.
/SUPPRESS	Trailing blanks are suppressed. The default is no suppression.
/TPUNCH	SPRINT-10 places the specified file into the paper-tape punch output queue. The default is that the file is not placed in the output queue.
/WIDTH:nn	Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks. If this switch is not specified or if no argument is specified, then SPRINT-10 will assume a value of 80.

Switch

Meaning

/026

The card deck is read in 026 card code. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the SJOB card. Refer to Paragraph 3.3.13.

#### Restrictions

The /BINARY, /026, and /IMAGE switches apply only to card reader input. Input from other devices must be read in ASCII code; otherwise, an error message is written in the log file and the job is terminated.

### 3.3.6 \$DUMP

#### Function

This card causes SPRINT-10 to insert a DUMP monitor command into the control file which invokes a dump when an error is detected by the Batch Controller.

Refer to Chapter 2 for description of the DUMP program.

#### Card Format

\$DUMP

### 3.3.7 \$EOD

#### Function

This card terminates the input that is being copied into a data file by SPRINT-10 because of a preceding File Card (refer to Paragraph 3.3). All control cards with the exception of \$MODE perform this action; i.e., terminate the copying of input. If input is not being copied and this card is read, SPRINT-10 ignores it. \$EOD is only necessary when the user wishes to place a line of input which is not a SPRINT-10 control card after input that is being copied into a data file.

Card Format

\$EOD

### 3.3.8 \$EOJ

Function

This card is the last card in a job deck and it terminates the job. The \$EOJ card has the same function as the standard end-of-file card. However, it is preferred to the end-of-file card when running a job from a remote batch station.

Card Format

\$EOJ

### 3.3.9 \$ERROR \$NOERROR

Function

- These cards are used to aid the Batch Controller in processing errors. They cause SPRINT-10 to insert an .IF statement into the control file; e.g., .IF (ERROR) or .IF (NOERROR). Refer to Paragraph 3.4.5 for an explanation of the .IF statement. These cards must appear at the point at which the error occurs.

Card Formats

\$ERROR statement

\$NOERROR statement

statement

= an executable monitor or batch command preceded by a period. If the statement directs the Batch Controller to go to a statement label, the statement label line and any related lines must be included in the sequence of commands at the place the user wants it executed. For example,

```
$FORTRAN TEST1  
.  
.  
.  
$ERROR .GOTO A  
$DATA TESTIDA
```

```
.  
. .  
$ERROR .GOTO A  
A::CONT  
$FORTRAN TEST2  
. .  
.
```

### 3.3.10 \$EXECUTE

#### Function

This card causes SPRINT-10 to insert an EXECUTE monitor command into the control file. It performs the same function as the \$DATA (Paragraph 3.3.4) only it does not have a data deck following it. The \$EXECUTE card is used when there is no data or when the data file already exists on disk (for example, through the previous use of a \$DECK card). The files to be placed in the EXECUTE command string are determined in the same way as they are for the \$DATA.

#### Card Format

\$EXECUTE/switch

/switch = /MAP

The only permissible switch on this card is /MAP which, when specified, causes a loader map to be generated and printed.

### 3.3.11 \$FORTRAN or \$F40

#### Function

This card precedes a FORTRAN source program and causes SPRINT-10 to copy the program onto disk and to insert a COMPILE monitor command into the control file. The source program is ended by any SPRINT-10 control card other than a \$MODE.

The \$FORTRAN card causes SPRINT-10 to invoke the FORTRAN-10 compiler and the \$F40, the F40 compiler.

When the job is run, the specified program is compiled and temporary relocatable binary and listing files are created. The source, binary and listing files can be made permanent if the user renames them to change their protection. (Refer to the RENAME command in Chapter 2.) The listing file is printed as part of the job's output.

The \$FORTRAN card does not cause execution of the program. Execution is initiated by an EXECUTE monitor command or by a \$DATA or a \$EXECUTE card. (Refer to the EXECUTE command in Chapter 2 or paragraphs 3.3.4 and 3.3.10 in this chapter respectively.)

Processor switches can be passed to the FORTRAN compiler by including them in the command string. Refer to Paragraph 1.5.7 for a description of the FORTRAN processor switches.

### Card Format

\$FORTRAN dev:name.ext [proj,prog] (processor switches) /S(1)/S(2).../S(n)

\$F40 dev:name.ext [proj,prog] (processor switches) /S(1)/S(2).../S(n)

- dev: = a file structure name. If omitted, DSK is assumed.
- name.ext = the name of the file to be created on disk. If omitted, SPRINT-10 assigns the filename LN???? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique filename) with the extension .FOR for FORTRAN-10 and .F4 for FORTRAN-40. However, when the user specifies only a filename (with no extension), SPRINT-10 will not append any extension. The source and relocatable binary files will be explicitly deleted at the end of the job if no filename was specified. Where the user has specified a filename, the disposition of the files depends on the protection the user has given them.
- [proj,prog] = a directory name other than that specified on the \$JOB card. If omitted, the project-programmer number on the \$JOB card is used.
- (processor switches) = the switches to be passed to the FORTRAN compiler. They must be enclosed in parentheses and the slash must not appear in connection with these switches.
- /S(1)/S(2).../S(n) = the switches that control the mode of input interpretation and the listing of the compiled program.

### Switch

### Meaning

/ASCII

The input is read in ASCII mode. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.

Switch	Meaning
/CREF	= A cross-referenced listing is created to be processed by the CREF program. The default is no listing.
/NOLIST	= No listing file of the program is created. If this switch is not specified, a listing file will be created.
/NOSUPPRESS	= Trailing blanks are not suppressed. The default is no suppression.
/SUPPRESS	= Trailing blanks are suppressed. The default is no suppression.
/WIDTH:nn	= Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks. If this switch is not specified or if no argument is specified, then SPRINT-10 will assume a value of 80.
/026	= The card deck is read in 026 card code. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13. This switch applies only to card-reader input. Input from other devices must be read in ASCII code; otherwise, an error message is written in the log file and the job is terminated.

### 3.3.12 \$INCLUDE

#### Function

This card causes already existing relocatable binary files to be loaded with the user's programs. A file specified on a \$INCLUDE card is added to the list of filenames remembered by SPRINT-10 and included in the EXECUTE command string generated by a \$DATA or by a \$EXECUTE card. Refer to Paragraphs 3.3.4 and 3.3.10.

#### Card Format

\$INCLUDE name.ext[proj,prog], /switch

name.ext	= the name of the file to be loaded. The extension is normally .REL. If the filename is omitted, an error message is issued and the job is terminated.
----------	--

[proj,prog] = a directory name other than that specified on the \$JOB card. If omitted, the project-programmer number on the \$JOB card is used.

Switch	Meaning
--------	---------

/SEARCH	This switch specifies that the file is to be loaded in library search mode.
---------	---

### Restrictions

The file specified on the \$INCLUDE card must be a relocatable binary file and must already exist on a disk.

### 3.3.13 \$JOB

#### Function

This card, in conjunction with the \$PASSWORD card (if required), causes SPRINT-10 to create a control file and a log file in the user's disk area into which commands are placed for the Batch Controller.

SPRINT-10 uses the name specified on the \$JOB card for the control and log files with the extensions .CTL and .LOG respectively. If the job-name is omitted, SPRINT-10 creates a unique name for the job. Refer to the Card Format section for a more detailed description. In general, the jobname used on input appears in the output queues.

The user may specify a wildcard designation (#) for the programmer number in the \$JOB card, for example,

\$JOB FLEX[4,#] or \$JOB FLEX(4,#) or \$JOB FLEX<4,#>

SPRINT-10 checks to determine if wildcard programmer numbers are permissible for the specified project. If they are, then SPRINT-10 creates a unique programmer number. If they are not allowed, then SPRINT-10 issues an error message and the job is terminated.

#### Card Format

\$JOB name [proj,prog] /S(1)/S(2).../S(n)

name	= the user-assigned name for the job; if omitted, SPRINT-10 creates the unique name JB???? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique name) for the control and log files.
------	---



[proj,prog] = the project-programmer number of the user who submitted the job. This argument is required. A space or comma can separate this argument from the job-name.

/S(1)/S(2).../S(n) = switches taken from the following group. These switches are optional.

Switch	Default	Meaning
/AFTER:dd-mmm-yy hh:mm	None	The job cannot be run until after the specified date and time. The resulting AFTER time must be less than the DEADLINE time.
/AFTER: + hh:mm	None	The job cannot be run until after the input time plus the amount of time specified.
/CARDS:nn	0	The maximum number of cards (up to 10,000) that can be punched by the job (in decimal).
/CORE:nnK	25K	The maximum amount of core (in decimal) that can be used by the job up to the maximum allowed by the installation. Specifying in K is optional.
or /CORE:nnP	50P	The maximum amount of core (in pages) that can be used by the job up to the maximum allowed by the installation.
/DEADLINE:dd-mmm-yy hh:mm	None	The job must be completed by the specified date and time. The resulting DEADLINE time must be greater than the AFTER time.
/DEADLINE: + hh:mm	None	The job must be started by the indicated amount of time after it is input.

Switch	Default	Meaning
/DEPEND:nn	0	Initial interjob dependency count (in decimal).
/ERROR:aa:bb:cc		Where:
	5	aa represents the number of binary checksum errors permitted before the job is terminated.
	5	bb represents the number of illegal binary cards permitted before the job is terminated.
	100	cc represents the number of Hollerith errors permitted before the job is terminated.
		If only one argument is specified, SPRINT-10 assumes it to be the value for cc. If two arguments are given, it is assumed that they are the values for bb and cc. The values not specified are assumed to be zero. For example:
		/ERROR:20
		means that the values for aa, bb and cc are 00, 00 and 20 and:
		/ERROR:15:20
		means that the values for aa, bb and cc are 00, 15 and 20. If the switch is not specified then the default values given above are in effect. However, if the switch is specified but no arguments are given, then aa, bb and cc are each assumed to have a value of zero.

Switch	Default	Meaning
/FEET:nn	0	The maximum number of feet of paper tape that will be punched by the job.
/HOLLERITH:aa	ASCII	This switch specifies the default Hollerith code for interpreting cards following a \$DATA, \$DECK or \$-language card. The legal values are ASCII and 026. On each of the aforementioned control cards, switches may be specified to change the interpretation mode for that deck. Unless a new mode of interpretation is specified, each subsequent control card encountered resets the input mode to that specified on the Hollerith switch.
/LOCATE:oo	The station where the job was input	Specifies the remote station (in octal) of the job and where the output is to be sent.
/NAME:aa	None	The user's name in up to 12 characters.
/NORESTART		The job is not restartable. If this switch is omitted, it is assumed that the job can be restarted. (See /RESTART below.)
/PAGES:nn	200	The maximum number of pages (in decimal) to be printed by the job, including the log file and compilation listing.
/PRIORITY:nn	10	The external priority of the job: the highest priority that can be specified is 62 (decimal).
/RESTART		In the event of a system crash, the job may be restarted by the Batch system. This is the default action.
/SEQUENCE:nn	0	The job's sequence number within the input queue.

Switch	Default	Meaning
/TIME:hh:mm:ss	5.0(5 mins.)	The limit placed on the amount of CPU time used by the job.
/TPLOT:mm	0	The maximum amount of plotter time that the job will use (in minutes).
/UNIQUE:n	1	If 0 is specified, more than one Batch job may be run concurrently using the specified directory. If 1 is specified, only one Batch job is run using the specified directory. If 2 is specified, the job is run in a unique SFD.

#### Requirements

The \$JOB card must immediately follow the \$SEQUENCE card, or be the first card if the \$SEQUENCE card is not required.

### 3.3.14 \$MACRO

#### Function

This card precedes a MACRO source program and causes SPRINT-10 to copy the program onto disk and to insert a COMPILE monitor command into the control file. The source program is ended by any SPRINT-10 control card other than a \$MODE.

When the job is run, the specified program is compiled and temporary relocatable binary and listing files are created. The source, binary and listing files can be made permanent if the user renames them to change their protection. (Refer to the RENAME command in Chapter 2.) The listing file is printed as part of the job's output.

The \$MACRO card does not cause execution of the program. Execution is initiated by an EXECUTE monitor command or by a \$DATA or a \$EXECUTE card. (Refer to the EXECUTE command in Chapter 2 or paragraphs 3.3.4 and 3.3.10 in this chapter, respectively.)

Processor switches can be passed to the MACRO compiler by including them in the command string. Refer to Paragraph 1.5.7 for a description of the MACRO processor switch.

#### Card Format

\$MACRO dev:name.ext [proj,prog] (processor switches) /S(1)/S(2).../S(n)

- dev: = a file structure name. If omitted, DSK is assumed.
- name.ext = the name of the file to be created on disk. If omitted, SPRINT-10 assigns the filename LN???? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique filename) with the extension .MAC. However, when the user specifies only a filename (with no extension), SPRINT-10 will not append any extension. The source and relocatable binary files will be explicitly deleted at the end of the job if no filename was specified. Where the user has specified a filename, the disposition of the files depends on the protection the user has given them.
- [proj,prog] = a directory name other than that specified on the \$JOB card. If omitted, the project-programmer number on the \$JOB card is used.
- (processor switches) = the switches to be passed to the MACRO assembler. They must be enclosed in parentheses and the slash must not appear in connection with these switches.
- /S(1)/S(2).../S(n) = the switches that control the mode of input interpretation and the listing of the assembled program.

Switch	Meaning
--------	---------

/ASCII	The input is read in ASCII mode. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.
--------	---

/CREF	A cross-referenced listing is created to be processed by the CREF program. The default is no listing.
-------	---

/NOLIST	No listing file of the program is created. If this switch is not specified, a listing file will be created.
---------	---

Switch	Meaning
/NOSUPPRESS	Trailing blanks are not suppressed. The default is no suppression.
/SUPPRESS	Trailing blanks are suppressed. The default is no suppression.
/WIDTH:nn	Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks. If this switch is not specified or if no argument is specified, then SPRINT-10 will assume a value of 80.
/026	The card deck is read in 026 card code. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13. This switch applies only to card-reader input. Input from other devices must be read in ASCII code; otherwise, an error message is written in the log file and the job is terminated.

### 3.3.15 \$MODE

#### Function

This card causes SPRINT-10 to change the mode in which it is interpreting the input stream. The \$MODE card can be placed anywhere after the \$PASSWORD card in the command sequence. All the switches take effect immediately after the \$MODE card is read. The /ASCII and /026 switches remain in effect until the next \$-language card is read but the effects of the /NOSUPPRESS, /SUPPRESS and /WIDTH switches are not terminated until either the end of the job or, until they are changed by another \$MODE card. The \$MODE

card is the only control card that does not terminate the copying of input preceded by a \$DECK card.

#### Card Format

\$MODE /S(1)/S(2).../S(n)

/S(1)/S(2).../S(n)

= switches that control the mode of reading and interpreting of the input media.

Switch

Meaning

/ASCII

The input is read in ASCII mode.

/NOSUPPRESS

Trailing blanks are not suppressed.

/SUPPRESS

Trailing blanks are suppressed.

/WIDTH:nn

Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks.

/026

The card deck is read in 026 card code.

#### NOTE

There are no defaults for switches on a \$MODE card because if a switch is not specified, the previously established value remains unchanged.

## Restrictions

- | The mode switch /026 can be used only for card input. Input from other devices is always read as ASCII code. Thus, the only switches that can be used with the \$MODE card for devices other than the card reader are /ASCII, /NOSUPPRESS, SUPPRESS and /WIDTH.

## | 3.3.16 \$PASSWORD

### Function

- | This card contains the password associated with the project-programmer number specified in the \$JOB card. If the password does not match the password stored in the system for the specified project-programmer number, SPRINT-10 does not create any files, issues an error message on the log file, and aborts the job. Use of this command is an installation option.

### Card Format

\$PASSWORD password

- | password = 1 to 6 character password. There must be exactly one space between the end of the card name (\$PASSWORD) and the first character of your password.

### Requirements

If the \$PASSWORD card is required, it must immediately follow the \$JOB card.

## | 3.3.17 \$RELOCATABLE

### Function

- | This is a File Card which causes SPRINT-10 to copy a relocatable binary program from cards to a file on the user's disk area. The cards are read in binary mode.

- | The difference between a \$RELOCATABLE card and the \$DECK card with the /BINARY switch specified is that all files created by a \$RELOCATABLE card are added to the list of files maintained by SPRINT-10 to be included in the EXECUTE command generated by a \$DATA or \$EXECUTE card. Refer to Paragraphs 3.3.4 and 3.3.10.

### Card Format

- \$RELOCATABLE name.ext



name.ext

= the name of the file into which the program is copied. If the filename is omitted, SPRINT-10 creates the filename RL???? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique filename). It is recommended that the user select a distinct name for each job in the Batch system simultaneously. If the extension is omitted, REL is assumed.

#### Restrictions

Relocatable binary programs can only be read when the input is from cards.

### 3.3.18 \$SEQUENCE

#### Function

This card specifies the job's unique sequence number. The use of this card depends on the requirements of the particular installation.

#### Card Format

\$SEQUENCE n

n

= a decimal number

#### Requirements

If the installation requires this command, it must be the first card in the input stream.

### 3.3.19 \$SNOBOL

#### Function

This card precedes a SNOBOL source program and causes SPRINT-10 to copy the program onto disk and to insert a COMPILE monitor command into the control file. The source program is ended by any SPRINT-10 control card other than a \$MODE.

When the job is run, the specified program is compiled and temporary relocatable binary and listing files are created. The source, binary and listing files can be made permanent if the user renames them to change their protection. (Refer to the RENAME command in Chapter 2.) The listing file is printed as part of the job's output.

Processor switches can be passed to the SNOBOL compiler by including them in the command string.

Refer to the COMPILE command in Chapter 2 for a description of the processor switches.

### Card Format

`$$SNOBOL dev:name.ext[proj,prog](processor switches)/S(1)/S(2).../S(n)`

- |                                   |  |
|-----------------------------------|--|
| <code>dev.</code>                 | = a file structure name. If omitted, DSK is assumed.   |
| <code>name.ext</code>             | = the name of the file to be created on disk. If omitted, SPRINT-10 assigns the filename LN???? (where ? represents a character arbitrarily chosen by SPRINT-10 to produce a unique filename) with the extension .SNO. However, when the user specifies only a filename (with no extension), SPRINT-10 will not append any extension. Then source and relocatable binary files will be explicitly deleted at the end of the job if no filename was specified. Where the user has specified a filename, the disposition of the files depends on the protection the user has given them. |
| <code>[proj,prog]</code>          | = a directory name other than that specified on the \$JOB card. If omitted, the project-programmer number on the \$JOB card is used.   |
| <code>(processor switches)</code> | = the switches to be passed to the SNOBOL compiler. They must be enclosed in parentheses and the slash must not appear in connection with these switches.  |
| <code>/S(1)/S(2).../S(n)</code>   | = the switches that control the mode of input interpretation and the listing of the compiled program.  |

Switch	Meaning
--------	---------

<code>/ASCII</code>	
---------------------	--

	The input is read in ASCII mode. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13.
--	---

Switch	Meaning
/NOSUPPRESS	Trailing blanks are not suppressed. The default is no suppression.
/SUPPRESS	Trailing blanks are not suppressed. The default is no suppression.
/WIDTH:nn	Columns 1 through nn (inclusive) are read. Any remaining columns are treated as if they contained blanks. If this switch is not specified or if no argument is specified, then SPRINT-10 will assume a value of 80.
/026	The card deck is read in 026 card code. If no other card format switch is specified, then the default is the mode indicated on the /HOLLERITH switch on the \$JOB card. Refer to Paragraph 3.3.13. This switch applies only to card-reader input. Input from other devices must be read in ASCII code; otherwise, an error message is written in the log file and the job is terminated.

The \$SNOBOL card is the only S-language card that does not generate an EXECUTE command through a \$DATA or \$EXECUTE card.

### 3.4 BATCH CONTROL FILE COMMANDS

Ordinarily, the Batch Controller reads the control file in a sequential manner. The commands described in this section can appear in the control file to interrupt the sequential processing of the control file in order to specify error recovery. If an error occurs in the job, the Batch Controller examines the next monitor level line in the control file for an .IF command to determine what action to take on the error. It does not search past the next executable

monitor line in the control file for the .IF command; therefore, if this command is used, it must be the next monitor command in the control file.

If the user does not wish to include an .IF command, he may include two types of error recovery routines in the control file, one type labeled %ERR (error processing for non-system programs) and the other labeled %CERR (error processing for compilers and system programs). A system program is one found on a device specified in the SYS search list. If SYS is assigned as a logical device name, the programs are considered user programs, not system programs. After an error occurs in the job and the next executable monitor line in the control file is not an .IF statement, the Batch Controller searches for the labeled error recovery control lines and processes the statements following these labels. These routines may be placed anywhere in the control file. Once the Batch Controller has processed the routine, it continues from that point in the control file; it does not read backwards over sections of the control file skipped in searching for error routines. The following example shows the use of a %ERR error recovery routine.

```
.COMPILE SAMPLE /LIST
.MOUNT MTA:3 /VID:42936
.EXECUTE
.DISMOUNT 3
.R SORT
*MUMP.SRT=FOR04.DAT/R80/K1.10
.QUEUE MUMP.SRT
%ERR:::CLOSE
.DUMP
.DISMOUNT 3
%FIN:::DELETE FOR04.DAT
```

Depending on the type of error found, the following operations are performed. If a compilation error occurs, only the compilation and the listing result. No tape is mounted. If an execution error results,

1. the program is compiled,
2. the tape is mounted,
3. the program begins execution,
4. the output is closed,
5. a quick dump of core is taken,
6. the tape is dismounted, and
7. the file FOR04.DAT is deleted.

If a SORT error occurs, the program compiles, the tape is mounted, the program is executed, and the file FOR04.DAT is deleted. Finally, if no errors result,

1. the program is compiled,
2. the tape is mounted,
3. the program is executed,
4. the tape is dismounted,
5. the sort is performed,
6. MUMP.SRT is printed, and
7. the file FOR04.DAT is deleted.

When the user is bypassing SPRINT-10 and creating his own control file, he may place a %FIN at the end of the control file. SPRINT-10, in creating the control file, automatically places a %FIN at the end. This label is used for cleanup purposes; e.g., deleting the input

files. In creating the control file, the user may place other %FIN's at various points in the file for periodic cleanup of his job. For example, this label is used in a special kind of error recovery. If the time allocated to the job runs to the maximum limit specified in the \$JOB command (refer to Paragraph 3.3.13) or by the Batch system, the user is given an additional 10% of his allocated time to clean up his job before it is aborted. Because the user includes a %FIN, cleanup is performed and the results of the job's processing are not lost when the job is aborted. The user should be careful in using the %FIN in the control file because if the Batch Controller is searching for an error recovery routine and %FIN is placed before a %ERR or %CERR, the %FIN is executed and the Batch Controller assumes the error recovery routine has been satisfied and does not search any longer for %ERR or %CERR. Furthermore, a .GOTO label cannot bypass a %FIN label. Therefore, the best place to put a %FIN is as the last line in the control file.

If an error occurs in the job and the user either was not running a system program or has not included an .IF command or error recovery control lines, the Batch Controller initiates a standard quick dump of the user's core area and terminates the job (refer to the DUMP command in Chapter 2). The Batch Controller also initiates a dump if it is searching for a %ERR and reads a %FIN instead.

### 3.4.1 .BACKTO

#### Function

The .BACKTO command is used by Batch users to interrupt the sequential reading of the control file by the Batch Controller. Control is transferred in a backward direction. This command can be used with a .IF command to specify transfer of control to an error routine.

#### Command Format

.BACKTO label

label

= label of a statement in the control file. This label is from one to six alphanumeric characters terminated with a double colon (::) and must not begin with a % character.

When the .BACKTO command is encountered, the Batch Controller searches for the labeled statement and transfers control to it. If the statement is not found, the job is terminated.

### 3.4.2 .CHKPNT

#### Function

The .CHKPNT command is used to aid in error recovery when a Batch job is terminated abnormally by a system failure. As many .CHKPNT commands as desired can be placed in the control file. When the job is restarted after the failure, the program begins at the location of the last .CHKPNT command instead of at the beginning of the program.

## Command Format

.CHKPNT label

label = label of the statement in the control file where processing is to resume when the job is restarted. This argument is required. The label is from one to five alphanumeric characters. When the label appears with the statement in the control file, it must be followed by a double colon (::).

### 3.4.3 .ERROR

#### Function

The .ERROR command causes the Batch Controller to recognize a message beginning with the specified character as an error in the job.

## Command Format

.ERROR character

character = the beginning character of the line that is to be recognized as an error (e.g., %). If this argument is not specified, a ? at the beginning of a line is considered as an error.

### 3.4.4 .GOTO

#### Function

The .GOTO command is used by Batch users to interrupt the sequential reading of the control file by the Batch Controller. Control is transferred in a forward direction. This command may be used with a .IF command to specify transfer of control to an error routine.

## Command Format

.GOTO label

label = label of a statement in the control file. The label appearing in the control file is from one to six alphanumeric characters terminated with a double colon (::) and must not begin with a % character.

When the .GOTO command is encountered, the Batch Controller searches for the labeled statement and transfers control to it. If the statement is not found before the end of the control file is reached, the job is terminated.

#### Examples

```
.EX TEST.MAC/L
.IF (ERROR) .GOTO A
.GOTO B
A::QUEUE LPT:=TEST.MAC

.GOTO B
B::;
```

### 3.4.5 .IF

#### Function

The .IF command is used by Batch users to aid the Batch Controller in processing errors. The Batch Controller recognizes the existence of an error when it encounters a line beginning with a question mark that is output from the job to the log file or a line that begins with the character specified in the .ERROR command. When the error occurs, this command must be the next monitor level in the control file.

#### Command Format

.IF (condition) statement

(condition) = ERROR or NOERROR. The parentheses must be included.

statement = an executable monitor or batch command preceded by a period.

If the specified condition is true, the statement is executed. If the specified condition is not true, the Batch Controller processes the next line in the control file.

### 3.4.6 .NOERROR

#### Function

The .NOERROR command instructs the Batch Controller to ignore all errors (including messages beginning with a question mark) in the job. This is especially useful in TECO searches. However, the message

## ?TIME LIMIT EXCEEDED

always indicates that an error exists.

### Command Format

```
.NOERROR
```

### 3.4.7 .NOOPERATOR

#### Function

.NOOPERATOR is the command to leave dialogue mode (Paragraph 3.1.3); it designates that no messages from the job are to be output to the controlling terminal. There is only one command that functions differently in dialogue mode; it is the .IF command. When in dialogue mode, upon encountering an .IF command, control is transferred to %ERR:: or %CERR:: if they exist or %FIN:: if it exists. If none of these are present, the job is automatically terminated. This action is taken in order to minimize output to the operator in case of an unexpected transfer of control. Therefore, in order to test for error conditions, it is necessary to issue a .NOOPERATOR command to leave dialogue mode before issuing an .IF command. For example:

```
.OPERATOR  
.RUN PRGTST  
.NOOPERATOR  
.IF(ERROR) .GOTO A
```

### Command Format

```
.NOOPERATOR
```

### 3.4.8 .OPERATOR

#### Function

The .OPERATOR command enters dialogue mode (Paragraph 3.1.3) and makes it possible for the job, or a program within the job, to communicate with the operator. Any message from the job, starting with the specified character (refer to Chapter 4), is typed on the controlling terminal. In addition, all subsequent lines, regardless of their initial character, are also copied to the operator. This process continues until the job requests input. At this point, rather than reading from the control file, the operator is notified of the input request and the job is suspended until the operator has responded. When the operator responds, the job is resumed and the response is sent to the job as if it had appeared in the control file. Also, copying is discontinued for all further output until another line beginning with the character specified is output. Then the process described is repeated.



When the .OPERATOR command is in effect, the Batch Controller ignores an .IF statement unless the .NOOPERATOR command is given first, and proceeds to search for an error recovery routine labeled with either %ERR:: or %CERR:: (refer to Paragraph 3.4). This action is taken in order to minimize output to the operator in case of an unexpected transfer of control. However, when an error occurs, the Batch Controller preserves the error status across the .NOOPERATOR command and looks for the .IF statement as the next monitor-level command. In other words, an .IF statement following a .NOOPERATOR command will be executed. Refer to the following examples.

In the example below, the .IF statement will be ignored.

```
.OPERATOR %  
.RUN TESPRG  
.IF (ERROR) .GOTO TAG
```

However, in the following example, the .IF statement will be executed.

```
.OPERATOR %  
.RUN TESPRG  
.NOOPERATOR  
.IF (ERROR) .GOTO TAG
```

#### Command Format

```
.OPERATOR character
```

character = the beginning character of the line that is to be sent to the operator (e.g., %). If this argument is not specified, \$ at the beginning of the line is assumed.

### 3.4.9 .REQUEUE

#### Function

The .REQUEUE command indicates to the Batch Controller that the job is to be requeued. It is normally used with the .if (error) command (e.g., .IF (ERROR) .REQUEUE). The job is restarted after a default requeue time at the specified label in the control file.

#### Command Format

```
.REQUEUE label
```

label = label of a statement in the control file where processing is to resume when the job is restarted. This label is from one to five alphanumeric characters. If this argument is omitted, the job is restarted at the beginning of the control file.

### 3.4.10 .REVIVE

#### Function

The .REVIVE command causes all output from the job to be placed in the log file.

#### Command Format

.REVIVE

### 3.4.11 .SILENCE

#### Function

The .SILENCE command suppresses all output from the job except error messages to the log file. This means that only the lines appearing in the log file will be those that begin with a question mark.

#### Command Format

.SILENCE

## 3.5 JOB OUTPUT

The output from a user's job is normally in the form of printed listings containing the user's job output, compilation listings, any memory dumps requested by the user or initiated by the Batch Controller, and the log file indicating the processing performed by the programs in the Batch system. The results from the job and the log file are automatically placed in the queue for the line printer spooler, LPTSPL, unless the job was submitted with the /OUTPUT:0 switch. However, the user can output to any device in the system. When a user program specifies a slow-speed spooling device, the Batch system places the output into a queue for the appropriate spooler. If the user wishes specific files to be output to particular spooled devices outside of his programs, he can so indicate on his \$DECK card (Paragraph 3.3.5) or he can include the QUEUE monitor commands in his control file to specify the output device and any additional parameters that he wishes.

Compilation listings are produced from the \$language control cards unless the user specifies otherwise. These listings are automatically spooled to the line printer. The user can also include the COMPILE monitor command in his job with switches to produce listings.

- ! The user can include any of the monitor DUMP commands or the SPRINT-10 card \$DUMP to request memory dumps during program testing. Under normal error conditions, the Batch Controller performs an automatic two-page dump for the user (refer to Paragraph 3.4).

### 3.5.1 The Log File

As part of its processing, SPRINT-10 creates a log file for each job so that the user can examine the processing performed by the Batch system. The log file contains a record of SPRINT-10's processing, control cards, any errors detected and any operator interventions.

When the job is run, the Batch Controller places additional messages into the log file, including each line of the control file as it is passed to the job, any error conditions, and any operator actions. The LOGOUT program appends an accounting summary message to the log file when the job terminates. This message is similar to the message received when an interactive user logs off the system (refer to the KJOB command in Chapter 2). Note that the log file is appended to for jobs of the same name; thus it may be necessary to delete this file before running another job with the same name.

**3.5.1.1 SPRINT-10 Messages** - SPRINT-10 places six kinds of messages into the log file. The first part of each message contains the time of day that SPRINT-10 placed the message into the file in columns 1 through 8 and an identifying word in columns 10 through 14. The body of the message is in columns 17 through 132. The identifier for each kind of message is taken from the following group:

STDAT -- gives the date, system name, SPRINT-10 version and the input device.

STMSG -- identifies any SPRINT-10 non-error message.

STERR -- identifies any SPRINT-10 error message.

STCRD -- identifies SPRINT-10 control cards.

STSUM -- identifies the summary message at the end of the job.

STOPR -- identifies any operator actions that occurred during SPRINT-10's processing.

The first entry in the log file always contains the identifier, STDAT and a message giving the date, the system name, the current version of SPRINT-10, and the input device; for example:

```
14:17:32  STDAT  23-Jan-74  5S06 SYS 40 SPRINT-10 Version 1 running on CDR0
```

The STDAT entry is followed by two STCRD lines containing the \$SEQUENCE and the \$JOB cards, respectively. The \$PASSWORD card is never printed for security reasons.

Each SPRINT-10 control card is written into the log on an STCRD line and, if it is a File Card, the line immediately following contains a message of the form:

```
14:17:40  STMSG file.ext Created - nn Cards read - nn Blocks written
```

When the \$EOJ card is read, SPRINT-10 prints a summary message giving the number of cards read, the number of files and blocks written, and the number of each type or error that occurred. The summary is also placed in the system accounting file. An example of the job summary is given below.

14:17:42 STSUM End-of-file after 423 cards  
14:17:42 STSUM 4 Hollerith errors  
14:17:43 STSUM Batch Input Request Created

Between the beginning and ending messages, SPRINT-10 prints any operator actions as they occur, some nonerror messages, and reports of errors it has detected. The following are examples of nonerror messages from SPRINT-10.

14:17:33 STCRD \$JOB TESTA, [10,225]  
14:17:33 STOPR [SPTSBO Job Stopped by Operator]  
14:17:33 STOPR [SPTCBO Job Continued by Operator]

**3.5.1.2 SPRINT-10 ERROR Reporting** - SPRINT-10 places messages in the log file that describe errors that have occurred during its processing. The following errors are detected, and their degree of severity is as specified:

#### Fatal Errors

- |  |   |   |
|--|---|---|
| <ol style="list-style-type: none"><li>1. Errors on the \$JOB or \$PASSWORD card.</li><li>2. Too many Hollerith errors.</li><li>3. Too many illegal binary cards.</li><li>4. Too many binary checksum errors.</li></ol> | } | Refer to the /ERROR switch on the \$JOB card - Paragraph 3.3.9. |
|--|---|---|

#### Non-fatal Errors

1. Hollerith errors.
2. Missing \$EOJ or end-of-file card.
3. Unrecognizable or illegal switch on a control card.

All SPRINT-10 error messages are preceded by the identifier STERR. The job is terminated upon encountering the first fatal error message. Any number of non-fatal errors are allowed unless limits are set by the user through the use of the /ERROR switch on the \$JOB card. (Refer to Paragraph 3.3.9.)

Refer to Chapter 4 for a complete list of SPRINT-10 error messages.

**3.5.1.3 Batch Controller Messages** - The times followed by an identifying notation are placed in columns 1 through 16 of the first line of each message. The identifiers for the Batch Controller messages are described in the list below:

BVERS -- denotes the version of BATCON.

BDATE -- identifies the date BATCON processed the job.

BATCH -- identifies any Batch Controller non-error message.

BAOPR -- describes any operator action.

BAERR -- denotes any Batch Controller error message.

MONTR -- identifies a line input or output at monitor level.

USER -- describes any line input or output at user level.

BASUM -- gives the Batch Controller summary message.

The first line in the log file printed by the Batch Controller is the version number. As each line in the control file is read, it is printed in the log file as well as being passed to the user program or to the monitor. Any time that the operator performs some action that affects the job, the Batch Controller records it in the log file. The BATCON program enters a message in the log file every time it generates a monitor command. For example, if a fatal error occurs in the job and the user has not included an .IF statement, a %ERR routine or a %CERR routine in the control file, the Batch Controller generates a DUMP command. It also generates a LOGIN and a KJOB monitor command for each job.

Any errors in the input that are detected by the Batch Controller are printed in the log file.

**3.5.1.4 Batch Controller Error Reporting -** The Batch Controller places the identifier BAERR on any line that it detects as being an error. The errors that are detected are listed below; the first three are fatal errors.

1. Missing condition (ERROR or NOERROR) or missing statement in an .IF statement.
2. Missing statement label in the .GOTO or .BACKTO command.
3. The labeled statement in a .GOTO command cannot be found after the .GOTO or before the .BACKTO command in the control file.
4. Use of the ATTACH, DETACH, SEND, CCONT, and CSTART monitor commands.

Most user error conditions are not flagged by the Batch Controller, they are passed to the monitor where they are flagged as errors.

## 3.6 SAMPLE JOBS

The following sample job setups illustrate the versatility of the Batch system.

■ The first example, Figure 3-2, shows a setup to list a card deck.

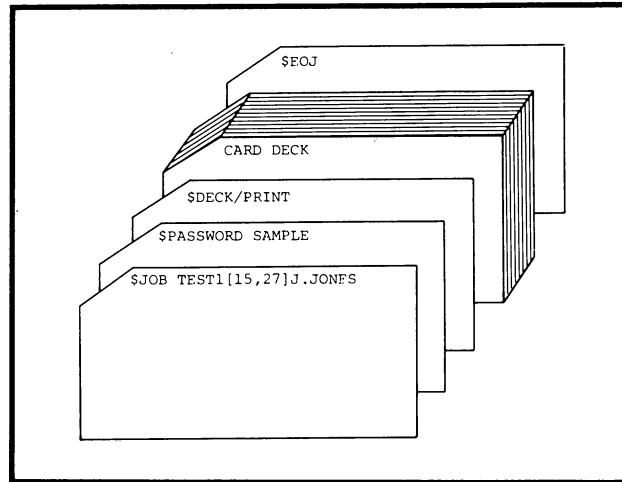


Figure 3-2 Sample Job #1

The second example, Figure 3-3, produces a CREF listing of a MACRO deck whether or not errors occur in the program.

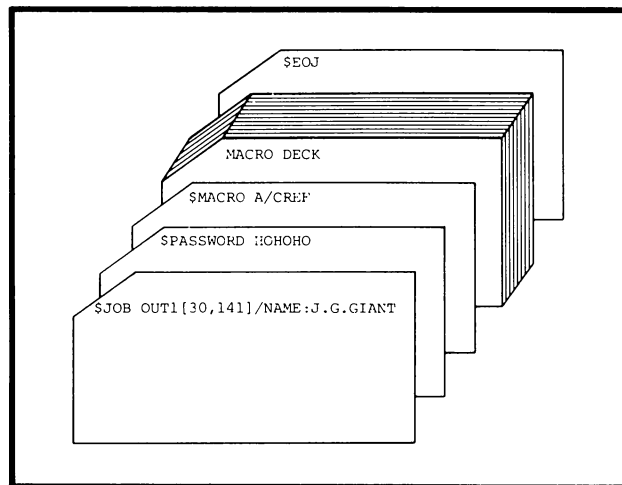


Figure 3-3 Sample Job #2

The third example, Figure 3-4, illustrates the use of error processing commands.

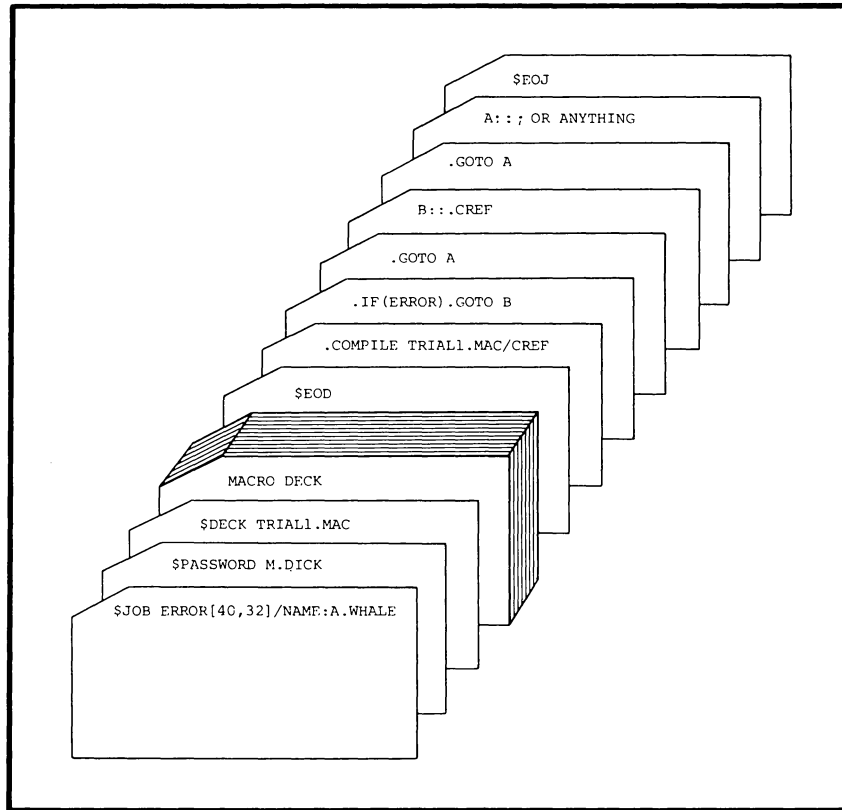


Figure 3-4 Sample Job #3

Figure 3-5 illustrates a MACRO assembly, two FORTRAN compilations, and execution of all three programs, and shows how monitor commands are entered along with the programs and the SPRINT-10 control cards.

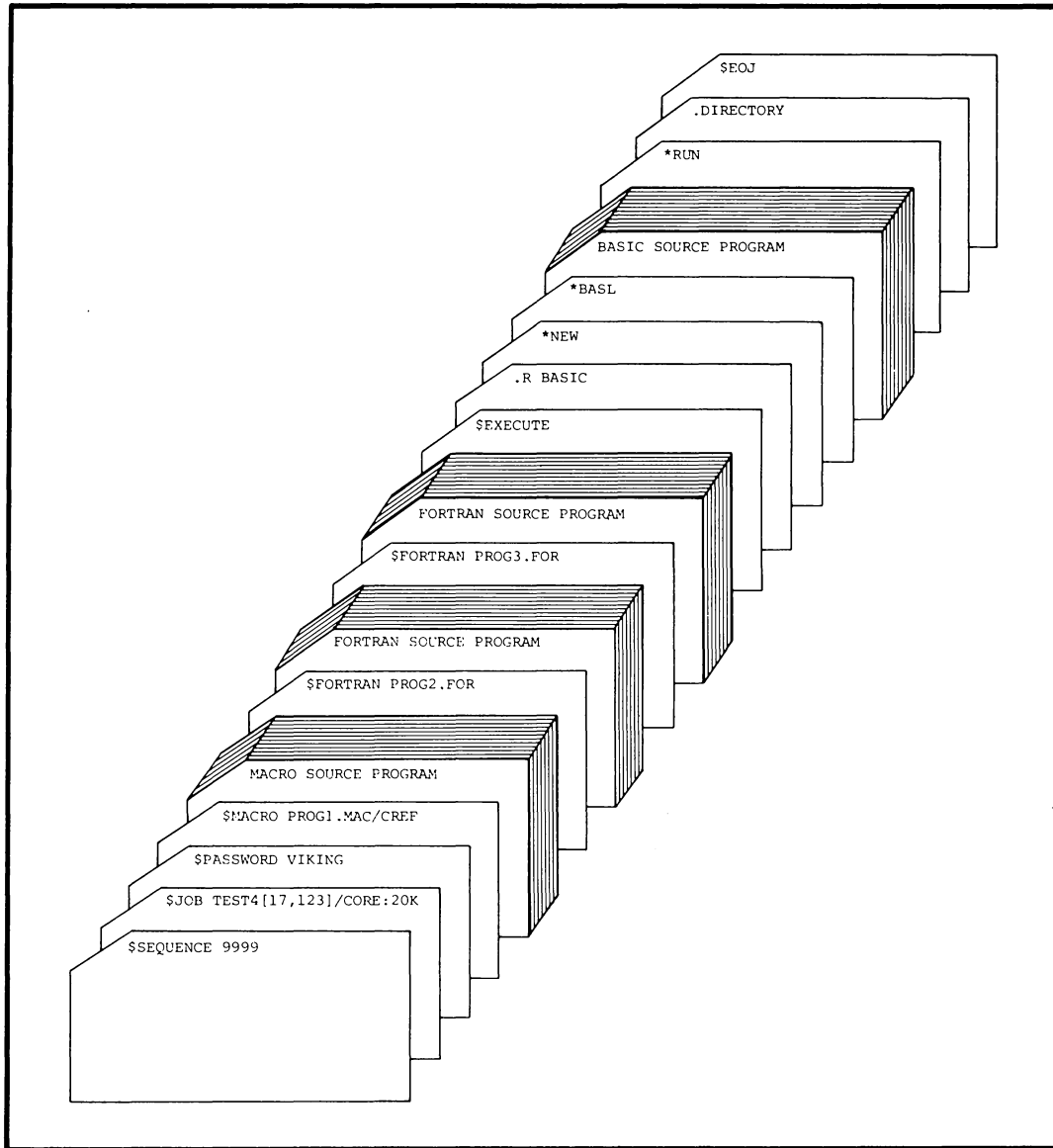


Figure 3-5 Sample Job #4



Figure 3-6 shows a simple SOUP update. Three base files are copied from cards to disk. The user files are on DECtape and the correction from DEC is on paper tape.

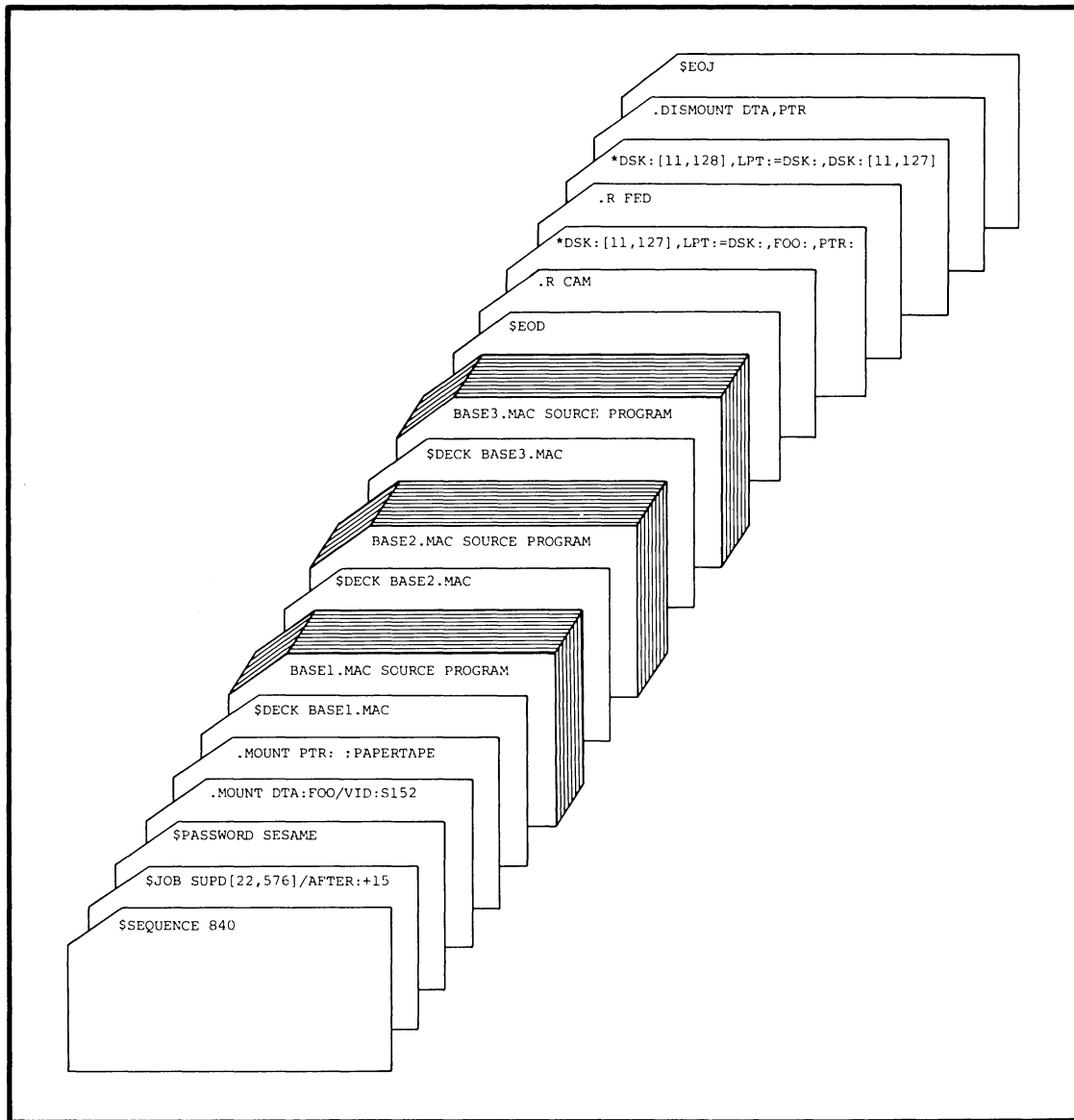


Figure 3-6 Sample Job #5

Figure 3-7 illustrates how to load a FORTRAN program with a System Library. The FORTRAN program will be compiled because of the \$FORTRAN card. The \$DATA card generates an EXECUTE monitor command and both the FORTRAN program and the relevant subroutines from SYS:SSP.REL will be included in the EXECUTE command string.

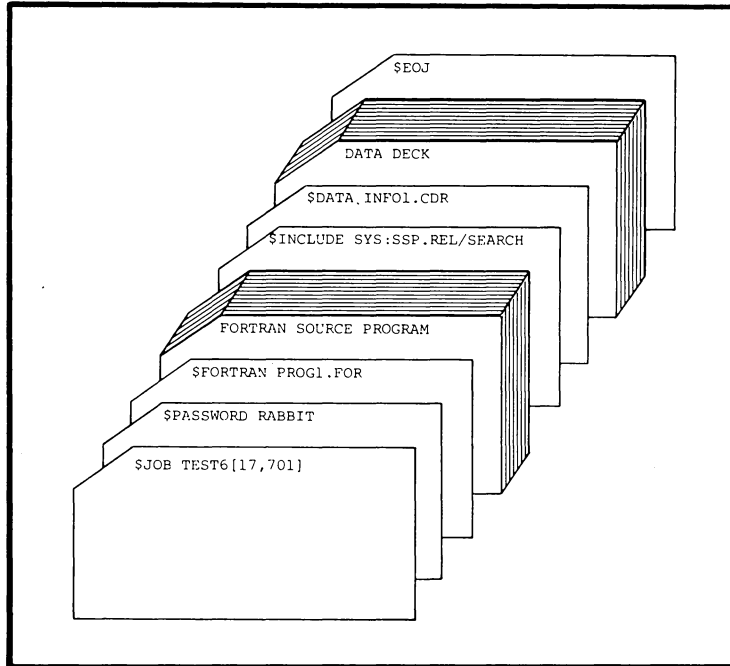


Figure 3-7 Sample Job #6

## CHAPTER 4

# SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

The following conventions are used in describing the system diagnostic messages:

dev	represents a legal device name.
file structure name	represents a legal file structure name.
file.ext	represents a legal filename and extension.
adr	represents a user address.
n	represents a number.
abc	represents a disk unit or drive.
x	represents an alphabetic character
switch	represents a switch.

Most messages returned to the user fall in one of five categories. These categories are determined by the beginning character of the message.

? at the start of the message indicates a fatal error message.

% at the start of the message represents an advisory or warning message.

[ at the beginning of the message indicates a comment line.

\$ at the beginning of the message represents an operator/job communication line. A response is expected.

“ (quote) at the beginning of the message represents a comment to the operator. No response is expected.

Programs and/or commands causing the error message are given in parentheses. However, all messages produced by SPRINT are preceded by one of the following sets of characters:

% SPT

? SPT

[ SPT

(Note that the ONCE-only messages have been removed from this section and placed in ONCE.RNO in the DECsystem-10 Software Notebooks.) The user can also employ the VERSION monitor command to determine the program that output the message. If the terminal is still in user mode (i.e., a character other than a period was output after the message), type the following

```
^C      (two ^C's if the program is not waiting for input)
.VERSION
```

The monitor returns with the name of the program in core (i.e., the one presumed to have output the message) and the version number of that program. After receiving the information type .CONT ↵ to return the terminal to user mode. If the message was a fatal message (i.e., a period was output after the message), ^C need not be typed since the terminal is already in monitor mode. In most cases, the user cannot type .CONT after a fatal error message.

The descriptive text given with the message indicates what action the user should take when he receives the message. He can, if necessary, notify the operator of any problems that he is having by issuing the SEND, PLEASE, or R GRIPE command.

#### 4.1 SYSTEM DIAGNOSTIC MESSAGES

The typein is typed back preceded and followed by ?.

The monitor encountered an incorrect character (e.g., a letter in a numeric argument). The incorrect character appears immediately before the second ?.

For example:

```
.CORE ABC
?CORE A?
```

#### ACCOUNTING SYSTEM FAILURE...

A program could not append an entry to the accounting file. Notify the operator. (LOGIN, LOGOUT).

#### ?ADDRESS CHECK FOR DEVICE dev

1. The monitor checked a user address on a UWO and found it to be too large (>C(JBREL)) or too small (<C(JBPF1)); in other words, the address lies outside the bounds of the user program
2. The SAVEd file is too large for the core assigned, or the file is not a core image file. (GET).

#### ?ALREADY ASSIGNED TO JOB n

The device is already assigned to another user's job (job n).

#### ?AMBIGUOUS ABBREVIATION

A command or switch has been abbreviated to the point that it is not unique. (COMPIL).

#### ?ARGS ARE: DAY, RUN, WAIT, READ, WRITE, VERSION, ALL, NONE

The user either did not type an argument or typed an illegal argument in the SET WATCH command string.

#### dev:ASSIGNED

The device has been successfully assigned to the user's job.

#### ?ASSIGNED TO JOB n(1), n(2), ...

If there is more than one device of the type specified, the numbers of the other jobs that have the same type of device are output, unless the user assigning the device has all the devices of the specified type. In this case, ?DEVICE ASSIGNED TO JOB is output.

#### ?ATTACH TO USER JOB FAILED

DAEMON could not attach to the user's job. (DAEMON).

#### ?BAD DIRECTORY FOR DEVICE DTAn

The system cannot read or write the DECTape directory without getting some kind of error. This error often occurs when the user tries to write on a write-locked tape or use a DECTape that has never been written on.

#### ?BATCH ONLY

The command issued can only be given by a batch job.

#### BLOCK NOT FREE

M specifies a unit or file structure logical block that is not free. (ALCFIL).

#### n BLOCKS ALREADY ALLOCATED

The file already exists. The new specification replaces, rather than updates, the old specification. (ALCFIL).

#### ?n1K BLOCKS OF CORE NEEDED

The user's current core allocation is less than the contents of .JBFF.

#### ?BOMB OUT

The location within INITIA that detected the error will be in AC 15 and the console lights. (INITIA).

#### ?BOOTSTRAP LOADER IS NOT IN COPY; TRY/L

An attempt was made to write the bootstrap loader onto a DEctape via the /T switch before the loader was loaded into a core buffer and preserved with the COPY core image. (COPY program).

#### ?BOOTSTRAP LOADER WILL NOT FIT IN 3 BLOCKS

The user's bootstrap loader is too big to fit into blocks 0, 1, and 2 of the output DEctape. (COPY program).

#### ?BUFFER CAPACITY EXCEEDED AND NO CORE AVAILABLE

The buffer is not large enough to handle the number of lines required for looking ahead for matches, and additional core is not available. (FILCOM).

#### ?BUSY

The terminal addressed is not communicating with the monitor (i.e., it is accepting a command or returning output from a command). The operator's terminal is never busy. (SEND, JCONT).

#### ?CANNOT DO I/O AS REQUESTED

Input (or output) cannot be performed on one of the devices specified for input (output). For example, input may have been requested for a device that can only do output. (FUDGE2).

#### ?CANNOT DO OUTPUT TO DEVICE dev

Output was attempted to a device that can only do input, or to a device assigned a logical name. (QUEUE).

#### ?CANNOT PROCESS EXTERNAL SYMBOLS

External symbols were encountered while loading the bootstrap loader with the /L switch. (COPY program).

?CANNOT PROCESS HIGH SEG'S

While loading the bootstrap loader with the /L switch, high segment code was encountered. (COPY program).

?CANNOT REATTACH FROM A BATCH SUBJOB

Batch jobs are not allowed to reattach their jobs. (REATTA).

?CANT ACCESS SYSTEM FILES

ACCT.SYS could not be read. Only the operator may LOGIN until ACCT.SYS is ready. Consult the operator. (LOGIN).

?CANT ADD TO YOUR FILE STRUCTURE SEARCH LIST n

n is the error code from STRUUO when trying to add a file structure to search list. (LOGIN).

?CANT ATT TO JOB

The project-programmer number specified is not that of the owner of the desired job, the project-programmer number was not given when it was required, or the PASSWORD given was incorrect. (ATTACH).

?dev CANT BE REASSIGNED

1. The job's controlling terminal cannot be reassigned, or
2. the logical name would be duplicated, or
3. the logical name is a physical device name in the system and the job reassigning the device is either logged-in under a different project-programmer number or does not have operator privileges. (REASSIGN).

?CANT CONTINUE

The job was terminated due to

1. all ERROR IN JOB messages (except for HALT),
2. the EXIT UUU,
3. the CLOSE command, or
4. the REA command when the device was INITed, and the user attempted to continue his program at the point at which I/O was terminated. The job cannot be continued.

## CANT CREATE NEW FILE STRUCTURE SEARCH LIST

The monitor cannot create a new file structure search list.

## ?CANT DECIPHER THAT

There is a syntax error in the command string. (MOUNT, DISMOUNT, FILE).

## ?CANT DET DEV

The user is not logged-in under [1,2].

## ?CANT ENTER OUTPUT FILE n file descriptor

The ENTER to write the output file failed; n is the disk error code. (DUMP).

## ?CANT EXPAND TABLE xxxx

The DUMP program ran out of core in attempting to expand the indicated table. (DUMP).

## ?CANT FIND INPUT FILE n file descriptor

DUMP cannot locate the file specified as the input file; n is the disk error code. (DUMP).

## ?CANT FIND FILE file.ext

The specified file could not be found.

## ?CANT GET SWAPPING PARAMETERS

DAEMON tried to obtain the job's swapping parameters and failed. (DAEMON).

## ?CANT GET SWAPPING POINTER FOR JOB

DAEMON tried to obtain the pointer to the user's job on the swapping space and could not because the GETTAB UO failed. (DAEMON).

## ?CANT GET USERS PPN

DAEMON tried to obtain the user's project-programmer number and could not because a GETTAB UO failed. (DAEMON).



?CANT OPEN file structure name

The file structure is mounted but cannot be opened. No UFD is created, though one may already exist. (LOGIN).

?CANT OPEN DEVICE dev

The specified device does not exist or it is assigned to another user. (DAEMON).

?CANT OPEN SWAP UNIT abc

DAEMON attempted to use the indicated swapping unit and failed. (DAEMON).

?CANT RENAME-FILE PRESERVED

An attempt was made via the /DISPOSE:RENAME switch to delete a preserved file (i.e., a file whose owner's field is greater than 0). (QUEUE).

?CANT SET OUR SEARCH LIST

DAEMON tried to set its search list and failed in its attempt. (DAEMON).

?CANT SET SEARCH LIST = USER'S

DAEMON attempted to set its file structure search list to be the same as the user's search list. (DAEMON).

?COMMAND ERROR

General catch-all error response for most commands. The syntax of the command is in error, and the command cannot be deciphered.

In FILCOM, one of the following errors occurred in the last command string typed.

1. There is no separator (← or =) between the output and input specifications.
2. The input specification is completely null.
3. The two input files are not separated by a comma.
4. A file descriptor consists of characters other than alphanumeric characters.
5. FILCOM does not recognize the specified switch.

6. The project-programmer number is not in standard format, i.e., [proj.prog].
7. The value of the specified switch is not octal.
8. The first input file is followed by a comma but the second input file is null.

#### ?COMMAND SWITCH REQUIRED

The given command string requires a FUDGE2 command code. (FUDGE2).

#### ?COMMAND SYNTAX ERROR TYPE /H FOR HELP

An illegal command string was entered. (GLOB).

#### ?COMMA REQUIRED IN DIRECTORY

A project-programmer number has been specified without the separating comma. (DUMP, QUEUE).

#### CONT BY OPR

The job has been continued by the operator. This message appears on the console of the job being continued. (JCONT).

#### ?CONTROL AND LOG FILES MUST BE DISTINCT

The control file cannot be the same file as the log file. (QUEUE).

#### ?2K CORE NEEDED AND NOT AVAILABLE

FILCOM needs 2K of core to initialize I/O devices and this core is not available from the monitor. (FILCOM).

#### %CPU<sub>n</sub> OPR1 ACTION REQUESTED

The Job's CPU specification includes a CPU which is not running or is not scheduling jobs. The monitor remembers the specification and uses the CPU as soon as it is started. If at least one CPU is running, the message is printed only once, since the job can run on another CPU.

#### ?DAEMON FILE MUST BE WRITTEN ON A DISK

The device specified was a nondisk device. (DAEMON).

#### ?DAEMON NOT RUNNING

The DAEMON program has not been initialized. It must be started by the operator to allow the DUMP and DCORE commands to operate. (DUMP, DCORE).

#### ?DETACH UUU FAILED

DAEMON could not detach itself from the TTY. Note that DAEMON does not detach itself if it is loaded with DDT. (DAEMON).

#### ?DATA ERROR ON DEVICE PTR

A read error has occurred on the paper-tape reader.

#### ?DESTINATION DEVICE ERROR

An I/O error occurred on the output device. (GLOB).

#### ?DEVICE CANT BE REASSIGNED

1. The job's controlling terminal cannot be reassigned,
2. the logical name would be duplicated, or
3. the logical name is a physical device name and the job reassigning the device is either logged in under a different project-programmer number or is not the operator.

#### ?DEVICE ERROR ON OUTPUT DEVICE

A write error has occurred on the output file. (FUDGE2).

#### ?DEVICE INIT FAILURE

The specified device has been assigned to another job or does not exist.

#### ?DEVICE MUST BE A DECTAPE

The only device that can be specified in the COPY command string is the DECTape.

?DEVICE NOT ASSIGNABLE

A non-privileged user cannot assign the requested device because it belongs to the restricted pool of devices. The user should try to assign the device with the MOUNT command. (ASSIGN).

?DEVICE NOT AVAILABLE

Specified device cannot be initialized because another user is using it or because it does not exist.

?DEVICE WILDCARD ILLEGAL

The wildcard construction cannot be used in the device specification. (DUMP, QUEUE).

?DIALOG MODE NOT SUPPORTED

The capability of interactive dialogue with the user has not been implemented. (QUEUE).

?DIRECTORY FULL ON OUTPUT DEVICE

There is no room in the file directory on the output device to add the updated file (nondisk devices only). (FUDGE2).

device name DISMOUNTED

The DISMOUNT command has completed.

?device name DISMOUNT INCOMPLETE

The DISMOUNT command was unsuccessful. In most cases, the reasons for failure have already been listed by nonerror messages.

DONT KNOW CTY LINE NUMBER

The DCORE command cannot be typed on CTY. (DAEMON).

?DOUBLE DEVICE ILLEGAL

Two device names appeared in a row without an intervening filename, or two colons appeared in a row, e.g., LPT:PTP: or DSKA ::FILEX. (DUMP, QUEUE).

?DOUBLE DIRECTORY ILLEGAL

Two directory names cannot appear without an intervening filename. (DUMP, QUEUE).

?DOUBLE EXTENSION ILLEGAL

Two extensions cannot appear without an intervening filename or comma. (DUMP, QUEUE).

?DOUBLE FILENAME ILLEGAL

Two filenames appeared in a row, or two periods appeared in a row; e.g., Q TEST1 TEST2 or TEXTX..MAC. (DUMP, QUEUE).

DPAn NO DRIVE AVAILABLE ON THIS CONTROLLER

The drives on the specified controller are all in use. (MOUNT).

?DSK CANT BE REASSIGNED

An attempt was made to reassign the prototype disk device data block (DDB).

?DSKCHR FAILURE n ON UNIT abc

The DSKCHR UUC gave an unexpected error return; n is the disk error code. Notify the operator. (DAEMON, KJOB).

?ENTER ERROR n  
?DIRECTORY FULL

No additional files can be added to the directory of the output device; n is the disk error code. (GLOB).

?ENTER FAILURE

The DECTape directory is full (i.e., there is no room for the file to be written on the DECTape).

?ENTER FAILURE n

The output filename is null; n is the error code for an illegal filename (nondisk devices only). (FUDGE2).

?ENTER FAILURE IN QUEUE MANAGER

QUEUE was unable to enter the files into the output queue. (QUEUE).

?ENTER FAILURE n ON DAEMON FILE

The ENTER to write the file failed; n is the disk error code.

?ENTRY BLOCK TOO LARGE PROGRAM name

The entry block of the named program is too large for the FUDGE2 entry table, which allows for 100 entry names. FUDGE2 can be reassembled with a larger table. (FUDGE2).

?ERROR CLOSING OUTPUT, STATUS = n

An I/O error occurred while closing the file on disk; n is the disk error code (DUMP).

?ERROR IN JOB 0

Usually this message indicates an error has occurred in the monitor. The message appears on either the user's terminal or the operator's terminal and precedes a one-line description of the error.

?ERROR IN DETACHED JOB n [prog name]

A fatal error occurred in the detached job or in the monitor while servicing the job. This message appears only on the operator's terminal (OPR) and precedes a one-line description of the error.

?EXCEED LOG-OUT m QUOTA BY n BLOCKS

The total number of blocks for all the user's files exceeds the maximum permitted value (m) by the indicated amount n. The user may use PIP or the DELETE command to remove files. Until the user is under the limit, he cannot dismount the file structure. (DISMOUNT).

?EXECUTION DELETED

A program is prevented from being executed because of errors detected during assembly, compilation, or loading. Loading is performed, but the loader exits to the monitor without starting execution. (LOADER).

?EXPECTED FORMAT IS 'NNNK' = 16K to 256K

The core-bank specified while processing the /T switch is not within the acceptable range or does not terminate with the letter K; e.g., 32 is not acceptable; 32K is. (COPY program).

file structure name FILE ERRORS EXIST

One of the files in a file structure has an error status, as flagged in the UFD of that file structure. (LOGIN).

?FILENAME ALREADY IN USE

The specified file already exists. (COMPIL).

?FILENAME REQUIRED FOR INPUT QUEUE

A file cannot be entered into the Batch input queue without a filename. (QUEUE).

?FILE NOT FOUND

File specified cannot be located if .LOW is not found in the same UFD as .SHR.

?FILE n NOT IN SAV FORMAT

The user indicated via the /X switch that the file is to be expanded but the specified file is not in compressed file format. N is either 1 or 2 indicating the first file or the second file. (FILCOM).

?FILE n READ ERROR

An error has occurred on either the first or second input device. (FILCOM).

%FILES ARE DIFFERENT

The two input files specified in the command string are different (i.e., the two files are not two versions of the same file but are two different files). (FILCOM).

?FILE SWITCHES ILLEGAL IN OUTPUT FILE

File switches cannot appear on the left of the equal sign, i.e., in the output specification. (QUEUE).

?(3) FILE WAS BEING MODIFIED-file.ext

Another user is modifying the file. (COMPIL).

?(0) FILE WAS NOT FOUND-file.ext

The named file could not be located. (COMPIL).

?FORMAT OR READ ERROR IN AUXACC.SYS

LOGIN unexpectedly found an end-of-file or an error in AUXACC.SYS. Notify the operator. (LOGIN).

file.ext FOUND BAD BY FAILSAFE READING MTA

The file in the file structure has an error status as flagged in the UFD of the file structure. (LOGIN).

FROM JOB n

An informative message telling the user the job number to which the console was attached or from which the console is detaching. (ATTACH, DETACH).

?FUDGE2 SYNTAX ERROR

An illegal command string was entered; for example, the left arrow was omitted or a program name was specified for the output file. (FUDGE2).

?GIVING BACK TOO MUCH CORE

An internal problem in the DUMP program. Notify your system programmer or software specialist. (DUMP).

?HALT AT USER adr

The user's program executed a HALT instruction at adr. Typing CONTINUE resumes execution at the effective address of the HALT instruction.

file.ext HARDWARE DATA READ ERROR DETECTED

The file has a hardware data read error flagged in the UFD of the file structure. (LOGIN).



file.ext HARDWARE DATA WRITE ERROR DETECTED

The file has a hardware data write error flagged in the UFD of the file structure. (LOGIN).

?HUNG DEVICE dev

If a device does not respond within a certain period after it is referenced, the system decides that the device is not functioning and outputs this message.

?ILLEGAL BLOCK TYPE dev:file.ext

The block type used is not in the range 0-77. (FUDGE2).

?ILLEGAL BLOCK TYPE

While loading the bootstrap loader with the /L switch, an unrecognizable block type was encountered by COPY. (COPY program).

?ILLEGAL COMMAND SYNTAX CHARACTER x

The character x is used incorrectly in the command string. (QUEUE).

?ILLEGAL DATA MODE FOR dev

The data mode specified for a device in the user's program is illegal, such as dump mode for the terminal. (FUDGE2).

?drive ILLEGAL DRIVE NAME

The drive specified by the user is in conflict with the unit or controller type required by the units of the file structure. (MOUNT).

?(0)ILLEGAL FILENAME

A filename of zero was specified. (FUDGE2).

?ILLEGAL IN BATCH JOB

The ATTACH, DETACH, SEND, CCONT, and CSTART monitor commands cannot be used by a batch job.

?ILLEGAL JOB NUMBER

The job number is too large or is not defined in this configuration.

?ILLEGAL QUEUE DEVICE

The queue name specified cannot be used with the given switch. (QUEUE).

?ILLEGAL QUEUE NAME xxx

The queue is not one of the system queues, or the queue is a logical name. (QUEUE).

?ILLEGAL SWITCH

A non-recognizable switch was used in the command string. (GLOB).

?ILLEGAL TO CREATE REQUEST FOR SOMEONE ELSE

Only the operator logged in under 1,2 can create queueing request for other users. (QUEUE).

?ILLEGAL UVO AT USER adr

An illegal UVO was executed at user location adr.

?ILL INST. AT USER adr

An illegal operation code was encountered in the user's program.

?ILL MEM REF AT USER adr

An illegal memory reference was made by the user's program. If this message occurred on a memory write, the error is at adr-1 since the program counter has been advanced. If it occurred on a memory read, then the illegal instruction is probably in location adr. The user should use the E command to first examine location adr-1 and then location adr in order to determine the illegal instruction. The index registers may also have to be examined.

?INPUT AND OUTPUT DECTAPES MAY NOT BE THE SAME DEVICE

The COPY program performs its operations on an input DECTape and an output DECTape. These DECTapes cannot be the same. (COPY program).

?INPUT (or OUTPUT) BLOCK TOO LARGE

A DECTape block number greater than 1101(octal) was encountered. (COPY program).

?INPUT (or OUTPUT) CHECKSUM OR PARITY ERROR

A read (or write) error has been detected. (COPY program).

?INPUT DEVICE dev CANNOT DO OUTPUT AT USER adr

Output was attempted on a device that can only do input (e.g., the card reader).

?INPUT (or OUTPUT) DEVICE ERROR

The DECTape control unit has detected the loss of data or a missed block. (COPY program).

?INPUT DEVICE NOT A DISK

The input specifications in a QUEUE command must be disk files. (QUEUE).

?INPUT ERROR

An I/O error occurred while reading a temporary command file from the disk. File should be rewritten. (COMPIL).

?INPUT ERROR - file.ext FILE NOT FOUND

The specified file could not be found on the input device. (FILCOM).

?INPUT ERROR ON DEVICE dev: STATUS (nnnnnn)

A data or device error occurred on input. (FUDGE2).

?INPUT ERROR, STATUS = n

An I/O error occurred while reading the file from disk; n is the disk error code. A new INPUT command causes a new LOOKUP to be done. (DUMP, DAEMON).

?INPUT (or OUTPUT) PREMATURE END OF FILE

When copying a DECTape, COPY encountered the end of file before it expected it. This may happen when copying a PDP-9 DECTape to a PDP-10 DECTape. (COPY program).

?INSUFFICIENT CORE FOR QUEUE

There is not enough core in system at the time of the KJOB command to make an output queue entry. (QUEUE).

?INVALID ENTRY - TRY AGAIN

#

An illegal project-programmer number or password was entered and did not match identification in system. The user is to retype his project-programmer number and password. (LOGIN).

?I/O TO UNASSIGNED CHANNEL AT USER adr

An attempt was made to do an OUTPUT, INPUT, OUT, or IN to a device that the user's program has not initialized.

#### CHARACTER

?x IS AN ILLEGAL SWITCH

An illegal character or switch was encountered in the command string. (FUDGE2).

?symbol IS A MULTIPLY DEFINED LOCAL

The named symbol is in more than one symbol table with different values. (DUMP).

?symbol IS AN UNDEFINED SYMBOL

The named symbol is not in DUMP's symbol table. (DUMP).

?symbol IS AN UNDEFINED SYMBOL TABLE NAME

The named symbol table has not been loaded with an XTRACT command. (DUMP).

?JOB CAPACITY EXCEEDED

1. This message is received by a user who attempts to login after the maximum number of jobs that the system has been set to handle has been initiated. The user should login at a later time. (LOGIN).
2. This message is received when a system program must create a job in order to perform its operation and the maximum number of jobs has been initiated. (SYSTAT, HELP, ATTACH).
3. This message may appear on rare occasions when an error made by the system hardware causes a reduction in the job capacity of the system.

?JOB NOT WAITING

The job specified is not waiting to be continued. (JCONT).

## JOB SAVED

The output is completed.

JOBn USER [p,p] LOGGED OFF TTY n AT hhmm dd-mm-yy  
DELETED <ALL> n FILES  
SAVED <ALL> n FILES m TOTAL BLOCKS USED  
ANOTHER JOB STILL LOGGED IN UNDER [p,p]  
RUNTIME n MIN m SEC

This information is typed as user logs off successfully. Note that m is total blocks allocated as opposed to blocks written. Therefore, it is always greater than or equal to the number of blocks written. Files are allocated in units of blocks called clusters. The system administrator selects the cluster size for each file structure, usually one block per cluster for FH file structures, and 5 or 10 blocks per cluster for DP file structures. (KJOB).

## ?LANGUAGE PROCESSOR CONFLICT

The use of the + construction has resulted in a mixture of source languages. (COMPIL).

## ?LEVEL D ONLY

The command issued is available only in 5-series monitors.

## ?LINKAGE ERROR - RUN UO

An I/O error occurred while reading a program from the device SYS:.  
(COMPIL).

## %LISTING DEVICE OUTPUT ERROR, STATUS =

The device specified for the output has an error. A new OUT command selecting a new file can be given or an OUT and APPEND command sequence to try again. (DUMP).

## ?LISTING ENTER FAILURE n

The ENTER to write the output file failed; n is the disk error code. (QUEUE).

## ?LISTING OPEN FAILURE ON DEVICE dev

The OPEN failed on device dev. (QUEUE).

## ?LOCKED-OUT BY OPERATOR

The operator is preventing any new accesses to the file structure in order that it may be removed. (MOUNT).

## file structure name LOGGED OUT QUOTA n EXCEEDED BY m BLOCKS

The user's allocation on the file structure named is greater than his logged out quota. The user must go through the CONFIRM dialogue and delete files until he is under the quota allowed to log off. (KJOB, LOGOUT).

## %LOGICAL NAME WAS IN USE, DEVICE dev ASSIGNED

The user previously assigned this logical name to another device. The logical name is cleared from the first device and assigned to the second.

## ?LOGIN PLEASE

A command that requires the user to be logged in has been typed to the monitor; it cannot be accepted until the user performs a LOGIN.

## ?LOGIN PLEASE TO USE SWITCH CREATE

The user must be logged in to make a new entry into a system queue. (QUEUE).

## ?LOOKUP ERROR n ?file.ext FILE NOT FOUND

The named file cannot be found in the directory on the specified device. (GLOB)

## ?LOOKUP FAILED, 'BSLDR.REL'

While processing the /L switch, COPY could not find the bootstrap loader named BSLDR.REL. (COPY program).

## ?LOOKUP FAILURE

The LOOKUP to read the disk file failed. This message is followed by a line explaining the reason for failure. (FUDGE2).

## ?file structure name LOOKUP FAILURE n

The LOOKUP to read the file failed; n is the disk error code.

?LOOKUP FAILURE FOR INPUT FILE n file

DUMP cannot read the input file. (DUMP).

?LOOKUP FAILURE n ON DAEMON FILE

The LOOKUP to read the DAEMON file failed; n is the disk error code. (DAEMON).

?MAX = n

A value was specified for an argument that is greater than the maximum value (n) allowed. (DUMP).

?MAY NOT LOGIN AS MFD PPN

No one can login as [1,1] because this number is the project-programmer number of the MFD. (LOGIN).

?MAY NOT LOGIN LOCAL  
REMOTE  
DATA SET  
BATCH JOB SUBJOB  
REMOTE CTY OR OPR

ACCT.SYS entry does not permit the project-programmer number to login at the terminal that is being used. (LOGIN).

?MAY NOT LOGOUT WITH FILE STRUCTURES FOR LOGICAL NAMES

A file structure in the job's search list is assigned a logical name, and only physical device names are recognized. The user should deassign the logical names. (KJOB, LOGOUT).

?MEM PAR ERROR AT USER PC adr

The processor detected a memory parity error in the low or high segment while the job was executing. The adr is the address of the PC stored by the hardware rather than the user address of the parity error. The operator also receives an error message giving the range of absolute addresses in case memory reconfiguration is necessary. DAEMON is awakened in order to record the pertinent information about the error for field service personnel.

The user must start a new copy of his program by typing the appropriate monitor command R, RUN, or GET. He should not start the program over by typing START, since the error is likely to reoccur or the program operate with incorrect data.

?MORE THAN ONE OUTPUT FILE ILLEGAL

Only one output queue-name may be specified in the QUEUE command string. (QUEUE).

?device MOUNTED

The device is mounted and ready for use. The MOUNT command has completed. If a file structure was mounted, a list of the unit ID's and the drives on which they are mounted is output. (MOUNT).

?device MOUNT INCOMPLETE

The MOUNT command has not completed successfully. In most cases, the reasons for failure have already been listed by nonerror messages. In a Batch job, MOUNT INCOMPLETE not preceded by a message may indicate that the user is attempting to mount a spooled device without executing a SET SPOOL command to unspool the device. The user must have unspool privileges in his accounting file entry in order to unspool and mount spooled devices.

?MUST BE IN OWNER'S PROJECT FOR SINGLE ACCESS

The user may not request single-access (/SINGLE switch) unless he has the same project number as the owner of the file structure. This requirement is enforced since a user with single access may execute super-USETI/USETO UUOs. (MOUNT).

name MUST NOT BE A LOGICAL NAME

The structure named contains the operator request queue (3,3.UFD) and must not be the logical name for some other structure. (MOUNT).

?file structure name MUST NOT BE WRITE-PROTECTED

The named structure is being used to queue requests to the operator and therefore may not be write-protected, SETSRC may be used to change the protection. (MOUNT, DISMOUNT, FILE).

NAME:

The ACCT.SYS entry for this project-programmer number requires the user to type a name which matches the one in ACCT.SYS in order to login. (LOGIN).

?NEED 5.03 OR LATER FOR REATTACH COMMAND

The REATTA program depends on UUOs available in the 5.03 release of the monitor. The user attempted to run the program using an older monitor. (REATTA).



?NESTING TOO DEEP

The @ construction exceeds a depth of nine and may be due to a loop of @ command files (COMPIL).

?NO CORE ASSIGNED

No core was allocated when the GET command was given and no core argument was specified in the GET.

NO DIFFERENCES ENCOUNTERED

No differences were found between the two input files. (FILCOM).

?(1) NO DIRECTORY FOR PROJECT-PROGRAMMER NUMBER - file.ext

A UFD does not exist for the requested project-programmer number. (COMPIL).

?NO END BLOCK ENCOUNTERED

The last block of the bootstrap loader program must be an end block (refer to the MACRO manual). (COPY program).

?NO ENTRY IN AUXACC.SYS

NO SEARCH LIST OR UFDS CREATED

If the user has no entry in AUXACC.SYS, LOGIN does not create UFDS or a search list. User is logged-in and has UFDS if they existed previously. He may write only on file structures that have UFDS or read all file structures. He may also create a file structure search list with SETSRC. The user can create UFDS on those file structures for which he has an entry in QUOTA.SYS by using the MOUNT command. (LOGIN).

NO ENTRY IN QUOTA.SYS

The user may utilize the file structure, but no UFD is created if he does already have one. (MOUNT).

%NO INFO ON 'name'

The user specified a feature that has no available documentation. (HELP).

?NO MODIFIER ALLOWED IN SWITCH switch

The switch specified cannot have an argument. (QUEUE).

## NONE PENDING

None of the user's requests to the operator are pending.

## ?NON-EXISTENT DRIVE DPAn

The user has specified a drive that does not exist in the system. (MOUNT).

## %NON-EXISTENT FILE input specification

The file specified for input could not be found. This message is not output if the /NEW switch is specified for the file. (QUEUE).

## ?NON-EX MEM AT USER adr

Usually due to an error in the monitor.

## ?NO OPR.JOB FOR THIS REQUEST

An operator request has been issued, but there is no OMOUNT running and enabled to service the request. The request is still queued unless the /PAUSE switch was given.

## ?NO PRIVILEGES TO SET CPU

The user does not have the privilege bits set by LOGIN from ACCT.SYS to change the CPU specification. The user should request that these privilege bits be set by the system manager.

## ?NO PRIVS TO SET PRIORITY THAT HIGH

The user does not have privileges to set such a high priority.

## ?NO PRIVS TO UNSPOOL

The user does not have privileges to unspool devices, and the operator has not set bit 28 in the STATES word.

## ?dev:file.ext<>NO PROGRAM NAME SPECIFIED

The switch (/D or /R) used in the command string requires that a program name be given. (FUDGE2).

?NO REMOTE USERS. TRY AGAIN LATER

The operator has used the SET SCHEDULE command to prevent LOGINs from remote terminals. The message of the day is still typed. (LOGIN).

NO ROOM IN QUEUE, TRY AGAIN LATER

There is no room in the queue for the user's request to be sent to the operator. (MOUNT).

?(14) NO ROOM OR QUOTA EXCEEDED - file.ext

There is no room on the file structure or the user's quota on the file structure has been exceeded.

%NO RUNNING CPUS IN SPECIFICATION

If none of the CPUs in the job's CPU specification are running, the user receives this message every minute until the CPU is started or he types a new SET CPU command.

?NO START ADR

Starting address or reenter address is zero, because the user failed to specify the starting address in the END statement of the source program or in the START command. However, an implicit starting address of 0 may be specified.

?NO SUCH DEVICE

The device name does not exist or was not assigned to this job.

?NO SUCH JOB

An attempt was made to attach to a job that has not been initialized.

?NO SUCH STR

A nonexistent file structure was specified. (KJOB).

?NO SUCH TTY

The terminal number is not part of the system configuration.

?NO SUCH UNIT

The unit does not exist or all units of this type are in use.

?NOT A JOB

The job number is not assigned to any currently running job. (ATTACH, DSK, JCONT). There is no job logged in at this terminal. (CONTINUE).

?NOT A SAVE FILE

The file is not a core image file.

?NOT A SPOOLING DEVICE

The device specified is not one of the spooling devices (LPT, CDP, CDR, PTP, PLT).

?NOT A STR - TRY AGAIN

The file structure specified is not recognized by the monitor.

?NOT A TTY

The device name given is not a terminal. (REATTA).

?drive NOT AVAILABLE

The drive indicated by the user is not currently available. (MOUNT).

?command NOT CODED

A command that is not in this version of DUMP was specified in the command string. (DUMP).

?NOT ENOUGH ARGUMENTS

An insufficient number of files of one type has been specified. (FUDGE2).

?NOT ENOUGH CORE

The system cannot supply enough core to use as buffers or to read in a system program. (COMPIL).

#### NOT ENOUGH DRIVES AVAILABLE

There are currently not enough drives of the right type to mount the file structure. (MOUNT).

#### NOT ENOUGH TABLE SPACE FOR SWAPPING UNITS

There are more swapping units than DAEMON allowed for. DAEMON should be reassembled. (DAEMON).

#### ?dev file.ext program NOT FOUND

The file or the program was not found on the device or in the file specified. If a program name is printed, this message may indicate that the program names in the command string appear in a sequence different from their sequence within the file. Therefore, the program may actually exist but was missed because of the incorrect sequence in the command string. (FUDGE2).

#### ?file.SAV NOT FOUND

The program file requested cannot be found on the system device or the specified device.

#### drive NOT READY

The indicated drive is either off-line or physically write-locked when write-enabled was requested. The operator will be notified. (MOUNT).

#### ?NOT YET SUPPORTED COMMAND CODE switch

A switch has been specified that is not implemented. (QUEUE).

#### NO UFD CREATED

The user may access the file structure, but he cannot write in his disk area since he has no UFD. (MOUNT).

#### ?NULL DEVICE ILLEGAL

A colon has been found without a preceding device name. (QUEUE).

#### ?NXM adr

While computing the value of an expression, a non-existent location was specified when referencing the input file. (DUMP).

?nk OF CORE NEEDED or ?nP OF CORE NEEDED

There is insufficient free core to load the files; n is the size being requested for the segment that failed (either high or low segment, not the sum of the high and low segments). This message occurs when the virtual core for the system has been exceeded or the core for this job has been exceeded. The user should type CORE ) to determine what core has been exceeded, and whether the high or low segment was too big. K denotes 1024 words which is the unit of core allocation on a KA10-based system, and P denotes 512 words (one page) which is the unit of allocation on a KI10-based system.

?OFFSET = 1000 TO 777600 (OCTAL)

The offset specified by the user is not within the acceptable range. (COPY program).

?ONLY BATCH USERS MAY LOGIN. TRY AGAIN LATER

The operator has used the SET SCHEDULE command to prevent LOGINs, except for BATCH jobs. The message of the day is still typed. (LOGIN).

?OPEN FAILURE ON DATA DEVICE dev

The OPEN on the specified device failed. (DUMP).

OPERATOR BUSY, HANG ON PLEASE.

The user must wait for the operator to become available.

OPERATOR NOTIFIED

1. The operator is available and the user may continue typing his message. (PLEASE).
2. A request is queued to the operator to perform a specified action. (MOUNT, DISMOUNT).

OPERATOR REQUESTED TO MOUNT UNITS

A request is queued to the operator to mount and ready the packs on the proper drives. (MOUNT).

OPERATOR REQUESTED TO READY DRIVES

One or more drives (as specified by previous messages) are not ready. A request is queued to the operator. (MOUNT).

#### OPERATOR REQUESTED TO REMOVE PACKS

A request to physically remove the packs has been queued to the operator. (DISMOUNT).

#### OTHER USERS - CANNOT SINGLE ACCESS

Other users are currently using the file structure that has been specified with the single-access switch (/SINGLE). The switch is ignored. (MOUNT).

#### OTHER USERS - CANT REMOVE

A DISMOUNT command requesting physical removal (/REMOV switch) of a pack has been issued and there are other users of the pack. The switch is ignored. (DISMOUNT).

#### OTHER USERS SAME PPN

A program has determined that other jobs are currently logged-in under the same project-programmer number. (LOGIN, KJOB).

#### ?OUT OF BOUNDS

The specified adr is not in the user's core area, or the high segment is write-protected and the user does not have privileges to the file that initialized the high segment. (D, E).

#### ?OUTPUT DEVICE dev CANNOT DO INPUT AT USER adr

An attempt was made to input from an output device (e.g., the line printer).

#### ?OUTPUT DEVICE ERROR

An error has occurred on the output device. (FILCOM).

#### ?OUTPUT ERROR

An I/O error occurred while writing a temporary command file on disk. (COMPIL).

#### ?OUTPUT ERROR, STATUS = n

An I/O error occurred while writing the file on disk; n is the disk error code. (DAEMON).

## ?OUTPUT INITIALIZATION ERROR

The output device cannot be initialized for one of the following reasons:

1. The device does not exist or is assigned to another job.
2. The device is not an output device.
3. The file cannot be placed on the output device. (FILCOM).

## PASSWORD:

The user must type a **PASSWORD** which matches that in the **ACCT.SYS** entry for this project-programmer number. Echoing is suppressed to preserve **PASSWORD** security. If the user is at a half-duplex (local copy) terminal, this message is replaced by a sequence of random over-typed characters, over which the user types his **PASSWORD**. (LOGIN).

## PAUSE...(^C TO QUIT, CR TO CONT)

The **/PAUSE** switch has been specified, and an operator action is about to be requested. **^C** aborts the command before the request is queued to the operator. Carriage return-line feed allows the command to continue, and the request is queued to the operator. (DISMOUNT).

## ?PC OUT OF BOUNDS AT USER adr

An illegal transfer has been made by the user program to user location **adr**.

## ?PLEASE KJOB OR DETACH

Attempt was made to **LOGIN** a job when the user already has a job initialized at that terminal. (LOGIN).

## ?PLEASE LOGIN AS [OPR]

The operator is the only person that can initialize **DAEMON** by typing **R DAEMON**.

## ?PLEASE TYPE ^C FIRST

A command which would start a job has been issued after a **CSTART** or **CCONT**.



#### ?PPN HAS EXPIRED

The current date is greater than the expiration date of the project-programmer number. The user may not login until expiration date is changed by the system manager. (LOGIN).

#### ?PROGRAM ERROR WHILE RESETTING MASTER DEVICE

FUDGE2 cannot find the master device or cannot find the program on the master device. (FUDGE2).

#### ?PROJECT 1 MAY NOT BE PTY

Project 1 is never allowed to login over a pseudo-TTY. (LOGIN).

#### ?(2) PROTECTION FAILURE - file.ext

There was a protection failure or the directory on DECTape had no room for the file. (COMPIL).

#### ?PTR INIT FAILURE

The logical device PTR is not available or could not otherwise be initialized. (COPY program).

#### QUOTA.SYS LOOKUP FAILURE

The LOOKUP to read QUOTA.SYS failed. (MOUNT).

#### QUOTA.SYS NOT ON STRUCTURE

QUOTA.SYS is not part of this structure. The user may still use the file structure, but no UFD will be created. (MOUNT).

#### QUOTA.SYS READ ERROR

An I/O error occurred while reading QUOTA.SYS. (MOUNT).

#### QUOTA.SYS WRONG FORMAT VERSION

Wrong version of QUOTA.SYS is on the file structure being mounted. Consult the operator. (MOUNT).

?file structure name RENAME FAILURE n

The RENAME to change the protection of the file failed; n is the disk error code. (KJOB, LOGOUT).

?(4) RENAME FILENAME ALREADY EXISTS - file.ext

The new filename on a RENAME command already exists. (COMPIL).

REQUEST STORED  
n COMMAND IN QUEUE

The request typed by the user has been placed in a queue to be performed when possible. n is the number of requests in the queue for all users. (FILE, MOUNT, DISMOUNT).

?REQUIRES DEVICE NAME

The device name or file structure name is required with the MOUNT and DISMOUNT commands.

?RIGHT BRACKET REQUIRED IN DIRECTORY

The project-programmer number must be enclosed in square brackets. (QUEUE).

%SEARCH LIST DOES NOT ALLOW CREATES

There are no file structures available to the user on which he can write. Run MOUNT or SETSRC to modify the search list as necessary. (LOGIN).

%SEARCH LIST IS EMPTY

There are no file structures in the DSK: search list that are available to the user. He can run the SETSRC program to modify his search list. (LOGIN).

?SINGLE-ACCESS BY JOB n

The file structure is already single access by the indicated user. (MOUNT).

file.ext SOFTWARE CHECKSUM OR REDUNDANCY ERROR

The file has no error as flagged in the UFD of the file structure. (LOGIN).

?SOME OTHER TIME

The user is not scheduled to LOGIN at this time. He should try again when he is allowed to login. (LOGIN).

?SORRY, CANT OPEN DSK, PLEASE CALL THE OPERATOR

This message is returned from the GRIPE program.

?SORRY, CANT WRITE IN COMPLAINT AREA, PLEASE CALL THE OPERATOR

This message is returned from the GRIPE program.

?SORRY, COMPLAINT BASKET IS FULL, PLEASE CALL THE OPERATOR

This message is returned from the GRIPE program.

?SORRY, NO UFD FOR COMPLAINT BASKET, PLEASE CALL THE OPERATOR.

This message is returned from the GRIPE program.

?SPTABO JOB ABORTED BY OPERATOR

The operator intentionally terminated the job.

?SPTAMO ASCII MODE ONLY ON NON-CDR DEVICES

The user has attempted to read a non-ASCII file on a device other than the card-reader.

%SPTBCK BINARY CHECKSUM ERROR ON CARD #n

A binary checksum error has been detected on card number n.

%SPTBDT BAD DATE-TIME SPECIFICATION ON /switch SWITCH - IGNORED

The specified switch will be ignored because the date-time was specified in an invalid or unrecognizable format.

[SPTCBO JOB CONTINUED BY OPERATOR]

The operator resumed the processing of the job. This message would occur subsequent to [SPTSBO JOB STOPPED BY OPERATOR].

?SPTCLB CAN'T LOGIN AS A BATCH JOB

The accounting system is set up so that this user is not permitted to run batch jobs. Consult the system administrator.

?SPTDDI DOUBLE DIRECTORY ILLEGAL

Two directory names cannot appear without an intervening filename.

?SPTDDV DOUBLE DEVICE ILLEGAL

Two device names appeared in a row without an intervening filename (e.g., LPT:PTP:) or two colons appeared in a row (e.g., DSKA::FILEX).

?SPTDEX DOUBLE EXTENSION ILLEGAL

Two extensions cannot appear in a row without an intervening filename or comma (.).

?SPTDFN DOUBLE FILENAME ILLEGAL

Two filenames appeared in a row (e.g., \$COBOL TEST1 TEST2) or two periods appeared in a row (e.g., TEST3..CBL).

?SPTDNA DEVICE dev NOT AVAILABLE

Specified output device cannot be initialized because it is currently being used or because it does not exist.

?SPTDND DEVICE dev IS NOT A DISK

The user asked to spool files onto a device that was not a disk.

?SPTENT ENTER ERROR n - xxx

Where n represents the number and xxx represents the text of one of the messages in Table 4-1 in section 4.2.

%SPTEPF EXTRANEIOUS PASSWORD CARD FOUND - IGNORED

Either the installation does not require a password, or the user included more than one SPASSWORD card with his job.

?SPTWF ERROR WRITING FILE

An I/O error occurred while writing the user's file.

#### ?SPTFSR FILESPEC REQUIRED ON \$INCLUDE CARD

The user has omitted the file specification on the \$INCLUDE card. It is necessary to correct the card and start the job again.

#### [SPTFST FILE SUBMITTED TO dev QUEUE]

Where dev represents the line-printer, plotter, card-punch or papertape.

#### %SPTHOL n HOLLERITH ERRORS IN CARD #n

n Hollerith errors were detected in card number n.

#### %SPTIBC ILLEGAL BINARY CARD - CARD #n

Card number n is not a legal binary card.

#### ?SPTICC ILLEGAL CONTROL CARD \$card name -- CARD #n

The card name on card number n is not a name recognized by SPRINT-10. Consult Chapter 3 for a description of the SPRINT-10 control cards.

#### ?SPTIDS ILLEGAL DIRECTORY SPECIFICATION

The format of the directory information enclosed within the square ([]) or angle (<>) brackets is incorrect.

#### ?SPTIFP ILLEGAL FORMAT FOR PPN

Refer to Chapter 3, section 3.3.13 for a description of the format of the project-programmer number.

#### ?SPTIPP INCORRECT PPN OR PASSWORD

Either the project-programmer number specified on the job card is not in the system accounting files or the password specified on the \$PASSWORD card is incorrect for the specified project -programmer number.

#### %SPTISV 'n' IS AN ILLEGAL VALUE FOR THE /switch SWITCH - VALUE IGNORED

This message warns the user that the argument he has used on the specified switch is not valid for that switch and that the job will be processed as though no argument had been specified. For example if the user had specified, /WIDTH: 999, SPRINT will assume /WIDTH:80. For more information, refer to Chapter 3.

**%SPTISW /switch IS NOT LEGAL ON THE \$card NAME CARD - IGNORED**

The switch specified by the user is not valid on the SPRINT control card. The switch will be ignored. Refer to the individual control card descriptions in Chapter 3 to determine the switches that are valid for each card.

**%SPTNAU NO AUXACC ENTRY**

There is no entry in the system file AUXACC.SYS for the project-programmer number specified on the \$JOB card. Consult the system administrator.

**?SPTNDV NULL DEVICE ILLEGAL**

A colon has been found without a preceding device name.

**%SPTNEF NO END-OF-FILE CARD FOUND - EOF ASSUMED**

SPRINT-10 encountered a new \$SEQUENCE or \$JOB without finding a previous \$EOJ or end-of-file card.

**%SPTNFT NO FILES TO LOAD**

The user has placed a \$DATA or \$EXECUTE card in his deck but has not preceded it by a \$-language, \$INCLUDE or a \$RELOCATABLE card. Refer to Chapter 3, sections 3.3.4 and 3.3.10.

**?SPTNYP SPECIFIED PPN IS NOT YOURS**

The user has attempted to run SPRINT-10 himself and the project-programmer number on the \$JOB card is different from the one he used to LOGIN.

**[SPTPRG UNIQUE PROGRAMMER NUMBER IS n]**

This message informs the user of the unique programmer number that LOGIN has assigned to him.

**?SPTPWR PASSWORD IS REQUIRED**

The user must give a password which matches the password in the ACCT.SYS for the specified project-programmer number.

**?SPTQTA QUOTA EXCEEDED ON dev WRITING USER FILE**

SPRINT was unable to spool the user's input due to the lack of sufficient space on the specified device.

[SPTSBO JOB STOPPED BY OPERATOR]

The operator temporarily stopped the job.

[SPTSFD UNIQUE SFD IS x]

The user has specified the /UNIQUE switch with a value of 2 on the \$JOB card.

?SPTSND SFD NESTING TOO DEEP

The directory specification in a filespec required a directory path which contained more than the maximum permissible number of SFD's.

?SPTTMB TOO MANY ILLEGAL BINARY CARDS

The number of illegal binary cards has exceeded the limit specified in the /ERROR switch on the \$JOB card.

?SPTTMC TOO MANY BINARY CHECKSUM ERRORS

The number of binary checksum errors has exceeded the limit specified in the /ERROR switch on the \$JOB card.

?SPTTMH TOO MANY HOLLERITH ERRORS

The number of Hollerith errors has exceeded the limit specified in the /ERROR switch on the \$JOB card.

%SPTUCO UNEXPECTED CHARACTER(S) 'x' ON CONTROL CARD - IGNORED

While scanning for a switch, SPRINT-10 encountered character 'x'. SPRINT-10 will ignore this unexpected character and will look for a slash (/), (indicating the next switch) or the end of card and ignore any information between. For example, if the user has accidentally punched or typed a question mark (?) instead of a slash (/) to indicate a switch, then that switch will be ignored.

%SPTURS /switch IS AN UNRECOGNIZED SWITCH - IGNORED

The switch specified is not recognized by SPRINT-10.

?STATION NOT IN CONTACT

The requested station is not in contact with the central station. (LOCATE).

?STATION NUMBER INVALID

The requested station number is not recognized by the system. (LOCATE).

STRUCTURE ALREADY MOUNTED

The requested file structure already exists and does not need to be physically mounted. (MOUNT).

?STRUCTURE NOT IN STRLST.SYS

The file structure name does not exist in the system administrator's file SYS: STRLST.SYS and, therefore, is not defined for the system. The operator or administrator may be requested to define the file structure by adding it to STRLST.SYS with the REACT program. (MOUNT).

?STRUUO FAILURE

The STRUUO UUO gave an error return. Notify the operator. (KJOB, LOGOUT).

%SUPERSEDING EXISTING FILE

A warning message indicating that a file already exists with the specified name. This file is being superseded. (TECO).

%SWAP READ ERROR UNIT abc STATUS = n

An I/O error occurred while reading the swapping space. The data is written into the DAEMON file as read. (DCORE).

?SWITCH ERROR

An illegal switch specification was given. (COPY program).

?switch SWITCH ILLEGAL

The switch specified cannot be used with the given queue name. (QUEUE).

?SWITCH VALUE TOO LARGE x

The value given to the switch exceeds the maximum value. (QUEUE).



#### ?SYNTAX ERROR

There is a syntax error in the command string. Check for incorrect parentheses or two operators in a row.

#### ?SYSSTR FAILURE

The SYSSTR UUO gave an error return. Notify the operator. (KJOB, LOGOUT).

#### ?SYSTEM ERROR - xxxxxx

System errors designate operator or system errors and are not a direct fault of the user. They are typed for possible diagnostic use.

#### ?SYSTEM NOT AVAILABLE

The operator has used the SET SCHED command to prevent LOGINs from timesharing terminals. The message of the day is still typed. (LOGIN).

#### ?TABLE OVERFLOW - CORE UUO FAILED TRYING TO EXPAND TO xxx

The GLOB program requested additional core from the monitor, but none was available. (GLOB).

#### ?THIS MONITOR WAS BUILT FOR A xxx AND WILL NOT RUN PROPERLY ON A yyy

The monitor is not running on the machine for which it was built. xxx and yyy are PDP-6, KA10, or KI10.

#### ?TIME LIMIT EXCEEDED

The time limit allocated for the job has been reached. The job is stopped and the terminal is returned to monitor mode.

#### TIMESHARING WILL CEASE IN m HOURS n MINUTES

The KSYS command (OPSER) or SET KSYS UUO has been issued in order to stop timesharing on the system at the indicated time.

#### ?TOO FEW ARGUMENTS

A command has been typed, but necessary arguments are missing.

#### ?TOO MANY FILENAMES OR PROGRAM NAMES

More than 40 program names or filenames were specified in the command string. The user should separate the job into several segments. (FUDGE2).

#### ?TOO MANY FILE STRUCTURES

The number of file structures exceeds the capacity of the monitor data base. The current limit is 14(decimal). (ONCE ONLY).

#### ?TOO MANY NAMES or ?TOO MANY SWITCHES

Command string complexity exceeds table space in the COMPIL program. (COMPIL).

#### ?TRANSMISSION ERROR

During a SAVE, GET, or RUN command, the system received parity errors from the device, or was unable to read the user's file in some other way. This can be as simple as trying to write on a write-locked tape.

#### ?TRANSMISSION ERROR ON INPUT DEVICE dev

A transmission error has occurred while reading data from the specified device. (FUDGE2).

#### ?TRIED TO OVERWRITE DATA WORD

After writing the core image file, DAEMON backs up to overwrite a word not known previously (e.g., the length of the category). In overwriting the word, DAEMON encountered a deviation from the standard pattern used in originally writing the word. (DAEMON).

#### ?TRY LARGER ARG

The specified argument is too small for the program. This message is followed by the standard output. (CORE).

#### ?TTYn ALREADY ATTACHED

Job number is erroneous and is attached to another console, or another user is attached to the job.

#### ?TTY IN USE

The terminal requested is already controlling a job or is otherwise in use. (REATTA).

## TYPE CORE BANK OR OFFSET FOR DTBOOT

On a /T switch, COPY asks for a core bank or offset for the bootstrap loader. The core bank is 16K to 256K and the offset is 1000 to 777600 (octal). (COPY program).

## TYPE H FOR HELP

An unintelligible response or command has been typed. Either the filename or the CONFIRM: message is repeated, depending upon what was typed. (KJOB).

## ?UFD ENTER FAILURE n

Failure in trying to create UFD; n is the disk error code. Notify the operator. (LOGIN).

## ?file structure name UFD INTERLOCK BUSY

Could not get UFD interlock when trying to set up a UFD. The UFD is not currently set up. Notify the operator. (LOGIN).

## ?UFD LOOKUP FAILURE n

A failure occurred in setting up a UFD; n is the disk error code. Notify the operator. (LOGIN).

## ?UFD OUTPUT FAILURE n

The output failed when trying to create the UFD (4-series); n is the software channel status. (LOGIN).

## ?file structure name UFD READ ERROR, STATUS = n

A read error occurred while reading the user's UFD on the file structure. Status n tells which error occurred. Notify the operator. (KJOB, LOGOUT).

## ?UFD RENAME FAILURE n

A failure occurred in setting up a UFD; n is the disk error code. Notify the operator. (LOGIN).

## ?UNDEFINED SWITCH switch

The specified switch is either undefined or not unique. (MOUNT, DISMOUNT).

#### ?UNEQUAL NUMBER OF MASTER AND TRANSACTION PROGRAMS

On a replace request, the number of master programs (or files) does not equal the number of transaction programs (or files). (FUDGE2).

#### UNIT abc ALREADY MOUNTED ON DRIVE DPAn

The file structure is already mounted but is on different drives than the user specified. (MOUNT).

#### ?UNKNOWN COMMAND

The monitor passed a command to COMPIL which COMPIL does not recognize. (COMPIL).

#### ?UNKNOWN DEFAULT FOR SWITCH switch

The default condition is not known for the specified switch. (DUMP, QUEUE).

#### ?UNKNOWN SWITCH switch

The switch named has been mistyped. (DUMP, QUEUE).

#### ?UNKNOWN SWITCH VALUE n

The argument specified with the switch has been mistyped. (DUMP, QUEUE).

#### ?UNRECOGNIZABLE SWITCH

An ambiguous or undefined word followed a slash. (COMPIL).

#### ?UUO AT USER adr

This message accompanies many error messages and indicates the location of the UUO that was the last instruction the user program executed before the error occurred.

#### n VERIFICATION ERRORS

On a word by word comparison requested via the /V switch, n discrepancies have been detected between the input DECTape and output DECTape. (COPY program).

#### WAITING...

A request has been queued to the operator and the command is waiting for the operator to complete the request. If the user does not want to wait for completion of the operator's action, he can type control-C without aborting the command. The operator action will still be completed. Later a DISMOUNT/CHECK or MOUNT/CHECK can be given to check for completion. (MOUNT, DISMOUNT).

#### WAIT PLS

The system's primary accounting file FACT.SYS was busy. It is retried for ten seconds before FACT.X01 is tried. This message can appear if many users are logging in simultaneously. (LOGIN, KJOB, LOGOUT).

#### %WARNING - INPUT REQUEST USES ONLY TWO ENTRIES

Only two files can be specified in the input queue request, the control file and the log file. (QUEUE).

#### !WARNING NO INDEX ON OUTPUT FILE-CONTINUING

The user has changed the structure of the index library file when deleting, appending, or inserting, thereby invalidating the index. The index has been removed from the new file. Reindexing is required. (FUDGE2).

#### ?dev WASNT ASSIGNED

The device is not currently assigned to the user's job and cannot be deassigned or reassigned by the job.

#### ?WASNT DET

The specified device is not detached.

#### ?WILDCARD ILLEGAL IN INPUT QUEUE FILE NAME DIRECTORY EXTENSION

The wildcard construction cannot be used when specifying the Batch input queue. (QUEUE).

#### ?WILDCARD ILLEGAL IN OUTPUT NAME DIRECTORY EXTENSION

The wildcard construction cannot be used in the output queue specification. (QUEUE).

?WRITE LOCK ERROR

An attempt was made to write on a write-locked DECtape. (COPY program).

?WRONG FORMAT FOR SYMBOL

A symbol was given in the format program :symbol and a symbol name did not follow the colon; in other words, the colon must be followed by a symbol. (DUMP).

?WRONG FORMAT VERSION NUMBER IN SYSTEM FILES

Wrong version of ACCT.SYS or AUXACC.SYS is on the system. Consult the operator so that he can run REACT to change the accounting files. (LOGIN).

YOU ARE LOGGED IN AS n,m

When a user logs in with a unique programmer number (project, #), this message informs him of the project-programmer number that LOGIN assigned. (LOGIN).

?YOU DONT HAVE PRIVILEGES TO WRITE           DAEMON  
  CCL       FILE

The user attempted to write in a file to which he did not have access. (DAEMON).

?1+1nK CORE  
VIR. CORE LEFT = 0

The swapping space or the core allocated to timesharing is all in use (i.e., there is no available virtual core). The user should wait a few minutes, and then attempt to login again. If this message still appears, it should be reported to the operator.

m+n/p CORE  
VIR. CORE LEFT = v

Key: m = number of blocks in low segment.

n = number of blocks in high segment.

p = maximum core per job. (Maximum physical user core unless limited by operator, or there are jobs locked in core (refer to DECsystem-10 Monitor Calls)).

v = number of K blocks unassigned in core and on the swapping device.

Note that nK represents 1024-word blocks which is the unit of core allocation on a KA10-based system, and nP represents 512-word blocks which is the unit of allocation on a KI10-based system.

## 4.2 ERROR CODES

The following error codes are returned in AC on RUN and GETSEG UUOs, in location E + 1 on 4-word argument blocks of LOOKUP, ENTER, and RENAME UUOs, and in the right half of location E + 3 on extended LOOKUP, ENTER, and RENAME UUOs. The codes are defined in the S.MAC monitor file.

Table 4-1

Error Codes

Symbol	Code	Explanation
ERFNF%	0	File not found, illegal filename (0,*), or file-names do not match (UPDATE).
ERIPP%	1	UFD does not exist on specified file structures. (Incorrect project-programmer number.)
ERPRT%	2	Protection failure or directory full on DTA.
ERFBM%	3	File being modified (ENTER).
ERAEF%	4	Already existing filename (RENAME) or different filename (ENTER after LOOKUP).
ERISU%	5	Illegal sequence of UUOs (RENAME with neither LOOKUP nor ENTER, LOOKUP after ENTER).
ERTRN%	6	One of the following errors occurred: <ol style="list-style-type: none"> <li>1. Transmission, device, or data error (RUN, GETSEG only).</li> <li>2. Hardware-detected device or data error detected while reading the UFD RIB or UFD data block.</li> <li>3. Software-detected data inconsistency error detected while reading the UFD or file RIB.</li> </ol>

Table 4-1 (Cont.)

## Error Codes

Symbol	Code	Explanation
ERNSF%	7	Not a saved file (RUN, GETSEG only).
ERNEC%	10	Not enough core (RUN, GETSEG only).
ERDNA%	11	Device not available (RUN, GETSEG only).
ERNSD%	12	No such device (RUN, GETSEG only).
ERILU%	13	Illegal UUO (GETSEG only). No two-register relocation capability.
ERNRM%	14	No room on this file structure or quota exceeded (over-drawn quota not considered).
ERWLK%	15	Write-lock error. Cannot write on file structure.
ERNET%	16	Not enough table space in free core of monitor.
ERPOA%	17	Partial allocation only.
ERBNF%	20	Block not free on allocated position.
ERNSD%	21	Cannot supersede an existing directory (ENTER).
ERDNE%	22	Cannot delete a non-empty directory (RENAME).
ERSNF%	23	Sub-directory not found (some SFD in the specified path was not found).
ERSLE%	24	Search list empty (LOOKUP or ENTER was performed on generic device DSK and the search list is empty).
ERLVL%	25	Cannot create a SFD nested deeper than the maximum allowed level of nesting.
ERNCE%	26	No file structure in the job's search list has both the no-create bit and the write-lock bit equal to zero and has the UFD or SFD specified by the default or explicit path (ENTER on generic device DSK only).
ERSNS%	27	A GETSEG from a locked low segment is not for a high segment that is a dormant, active, or idle segment.



# APPENDIX A

## STANDARD SYSTEM NAMES

### A.1 FILENAME EXTENSIONS

Table A-1 lists the filename extensions that have specific meanings in the DECsystem-10.

Table A-1

Filename Extensions

Filename Extension	Type of File	Meaning
ABS	Object	Absolute (nonrelocatable) program
AID	Source	Source file in AID language
ALG	Source	Source file in ALGOL language
ALP	ASCII	Printer forms alignment
ATO	ASCII	OPSER automatic command file
AWT	Binary	Data for automatic wire tester.
B10	Source	Source file in BLIS10
B11	Source	Source file in BLISS-11
BAC	Object	Reserved for output from the BASIC Compiler
BAK	Source	Backup file from TECO or LINED
BAS	Source	Source file in BASIC language
BCM	ASCII	Listing file created by FILCOM (binary compare)
BCP	Source	Source file in BCPL language
BIN	Binary	Binary file
BLB	ASCII	Blurb file

Table A-1 (Cont.)

## Filename Extensions

Filename Extension	Type of File	Meaning
BLI	Source	Source file in BLISS language
BUG	Object	Saved to show a program error
BWR	ASCII	Beware file listing warnings about a file or program
CAL	Object	CAL data and program files
CBL	Source	Source file in COBOL language
CCL	ASCII	Alternate convention for command file (@ command file construction for programs other than COMPIL)
CCO	ASCII	Listing of modifications to non resident software
CDP	ASCII, Binary	Spooled output for card punch
CFC	ASCII	Compressed file compare. Group of .SCM files combined with PIP.
CKP	Binary	Checkpoint core image file created by COBOL operating system
CHN	Object	CHAIN file
CMD	ASCII	Command file for indirect commands (@ construction for COMPIL)
CMP	ASCII	Complaint file by GRIPE
COR	ASCII	Correction file for SOUP
CRF	ASCII	CREF (cross-reference) input file
CTL	ASCII	MP batch control file
DAE	Binary	Default output for DAEMON-taken core dumps
DAT	ASCII, Binary	Data (FORTRAN) file

Table A-1 (Cont.)

Filename Extensions

Filename Extension	Type of File	Meaning
DDT	ASCII	Input file to FILDDT
DCT	ASCII	Dictionary of words
DIR	ASCII	Directory from FILE command or DIRECT program
DMP	ASCII	COBOL compiler dump file
DOC	ASCII	Listing of modifications to the most recent version of the software
DRW	Binary	Drawing for VB10C drawing system
DSE	ASCII	Directory sorted by extension
DSF	ASCII	Directory sorted by filename
ERR	ASCII	Error message file
F4	Source	Source file in F40 (FORTRAN) language
FAI	Source	Source file in FAIL language
FCL	Source	Source file in FOCAL language
FFS	ASCII	Fast FORTRAN stream
FLO	ASCII	English language flowchart
FOR	Source	Source file in FORTRAN-10 language
FRM	ASCII	Blank form for handwritten records
FTP	Source	FORTRAN test programs
FUD	ASCII	FUDGE2 listing output
GND	ASCII	List of ground pins for automatic wirewrap.

Table A-1 (Cont.)

## Filename Extensions

Filename Extension	Type of File	Meaning
HGH	Object	Nonsharable high segment of a two-segment program (created by SAVE command)
HLP	ASCII	Help files containing switch explanations, etc.
IDA	ASCII, Binary	COBOL ISAM data file
IDX	ASCII,SIXBIT	Index file of a COBOL ISAM file
INI	ASCII, Binary	Initialization file
LAP	ASCII	Output from the LISP compiler
LIB	ASCII	COBOL source library
LOG	ASCII	MPB or LINK-10 log file
LOW	Object	Low segment of a two-segment program (created by SAVE command)
LPT	ASCII	Spooled output for line printer
LSD	ASCII	Default output for DUMP program
LSP	Source	Source file in LISP language
LSQ	ASCII	Queue listing created by QUEUE program
LST	ASCII	Listing data created by assemblers and compilers
MAC	Source	Source file in MACRO language
MAN	ASCII	Manual (documentation) file
MAP	ASCII	Loader or LINK-10 map file
MEM	ASCII	Memorandum file
MID	Source	Source file in MIDAS (MIT Assembler) language

Table A-1 (Cont.)

## Filename Extensions

Filename Extension	Type of File	Meaning
MIM	Binary	Snapshot of MIMIC simulator
MSB	Object	Music compiler binary output
MUS	Source	Music compiler input
N	Source	Source file in NELIAC language
NEW	All	New version of a program or file
OBJ	Object	PDP-11 relocatable binary file
OLD	Source, Object	Backup source program
OPR	ASCII	Installation and assembly instructions
OVR	Object	COBOL overlay file
PAK	ASCII	Files compressed by PACK.TEC to save disk space
PAL	Source	Source file in PAL 10 (PDP-8 assembler)
P11	Source	Source program in MACX11 language
PL1	Source	Source file in PL1 language
PLO	Binary	Compressed plot output
PLT	ASCII	Spoiled output for plotter
PPL	Source	Source file in PPL language
PRO	Object	Program (save file)
PTP	ASCII, Binary	Spoiled output for paper-tape punch
Qxx	ASCII	Edit backup file, like .BAK (all xx)
QUD	ASCII, Binary	Queued data file
QUE	Binary	Queue request file

Table A-1 (Cont.)

## Filename Extensions

Filename Extension	Type of File	Meaning
QUF	Binary	Master queue and request file
REL	Object	Relocatable binary file
RIM	Object	RIM loader file
RMT	Object	Read-in mode (RIM) format file (PIP)
RNC	ASCII	RUNOFF input for producing a .CCO file
RND	ASCII	RUNOFF input for producing a .DOC file
RNO	ASCII	Programming specifications in RUNOFF input
RNP	ASCII	RUNOFF input for producing a .OPR file
RSP	ASCII	Script response time log file
RSX	All	Files for RSX-11D
RTB	Object	Read-in mode (RIM10B) format file (PIP)
SAI	Source	Source file in SAIL language
SAV	Object	Low segment from a one-segment program (created by SAVE command)
SCD	ASCII	Differences in directory
SCM	ASCII	Listing file created by FILCOM (source compare)
SCP	ASCII	SCRIPT control file
SEQ	ASCII, SIXBIT	Sequential COBOL data file, input to ISAM program
SFD	Binary	Subfile directory (reserved usage)

Table A-1 (Cont.)

## Filename Extensions

Filename Extension	Type of File	Meaning
SHR	Object	Sharable high segment file of a two-segment program (created by SAVE command)
SMP	Source	Source file in SIMPLE language
SNO	Source	Source file in SNOBOL language
SNP	ASCII	Snapshot of disk by DSKLST
SPT	ASCII	SPRINT - created files
SRC	ASCII	Source files
SVE	Object	.SAVed file from a single user monitor
SYM	Binary	LINK-10 symbol file
SYS	Binary	Special system files
TEC	ASCII	TECO macro
TEM	ASCII, Binary	Temporary files
TMP	ASCII, Binary	Temporary files
TST	All	Test data
TXT	ASCII	Text file
UFD	Binary	User file directory (reserved usage)
UPD	ASCII	Updates flagged in margin (FILCOM)
VMX	Object	Expanded save file starting at a location greater than zero and used as a special support program for virtual memory.
WCH	ASCII	SCRIPT monitor (WATCH) file
WRL	ASCII	Wirelist
XOR	Binary	Module data for XOR tester

Table A-1 (Cont.)

Filename Extensions

Filename Extension	Type of File	Meaning
XPN	Object	Expanded save file (FILEX and LINK-10)
Zxx	ASCII	Edit original file (all xx)

**A.2 RESERVED PROJECT-PROGRAMMER NUMBERS**

Table A-2 itemizes the project-programmer numbers that are allocated for specific functions in the DECsystem-10.

Table A-2

Project-Programmer Numbers

Number	Meaning
1,1	Master File Directory (MFD)
1,2	Operator functions
1,3	Old or superseded versions of system programs (device OLD:)
1,4	System library (device SYS:)
1,5	New or experimental versions of system programs (device NEW:)
1,6	PUB:
2,*	Recommended for operator's use
2.5	Storage for help text files (*.HLP) (device HLP:)
3,3	System and Multiprogram Batch (MPB) queues
4,*	Test and performance analysis systems



Table A-2 (Cont.)

## Project-Programmer Numbers

Number	Meaning
4,4	FAILSAFE testing
4,5	FAILSAFE testing
5,*	Libraries
5,1	BASIC source library (device BAS:)
5,2	COBOL source library for COPY verb (device COB:)
5,3	PDP-11 source library (device MXI:)
5,4	ALGOL source library (device ALG:)
5,5	BLISS source library (device BLI:)
5,6	FORTRAN source library (device FOR:)
5,7	MACRO source library (device MAC:)
5,10	Text editor library (device TED:)
5,11	Rel file library (device REL:)
5,12	RUNOFF library (device RNO:)
5,13	SNOBOL library (device SNO:)
5,14	Doc file library (device DOC:)
5,15	FAIL library (device FAI:)
5,16	MUSIC library (device MUS:)
5,17	MACRO universal files (device UNV:)
6,*	Field service and hardware diagnostics
7,7	Software acceptance
10,1	Special system programming storage region containing copies of SYS:CRASH.SAV. (device XPN:)
10,6	Software distribution
10,7	Software distribution (device DEC:)

### A.3 RESERVED DEVICE NAMES

Table A-3 lists logical device names (ersatz devices) that are predefined in the DECsystem-10.

To avoid confusion, it is recommended that these names not be used for private file structures.

Table A-3  
Device Names

Name	Use	UFD	Search List
ALG:	ALGOL library	[5,4]	System
ALL:		User's	All currently mounted structures
BAS:	BASIC library	[5,1]	System
BLI:	BLISS library	[5,5]	System
COB:	COBOL library	[5,2]	System
DMP	DUMP command files	[5,11]	System
DOC:	DOC file library	[5,14]	System
DSK:		User's	Job
FAI:	FAIL library	[5,15]	System
FOR:	FORTRAN library	[5,6]	System
HLP:	HELP library	[2,5]	System
LIB:	User defined library	Set by each user	System
MAC:	MACRO library	[5,7]	System
MUS:	MUSIC library	[5,16]	System
MXI:	PDP-11 library	[5,3]	System
NEW:	New system library	[1,5]	System
OLD:	Old system library	[1,3]	System

Table A-3 (Cont.)

Device Names

Name	Use	UFD	Search List
REL:	Rel file library	[5,11]	System
RNO:	RUNOFF library	[5,12]	System
SNO:	SNOBOL library	[5,13]	System
SYS:	System library	[1,4]	System
TED:	Text editor library	[5,10]	System
UNV:	MACRO universal library	[5,17]	System



## APPENDIX B CARD CODES

Table B-1

ASCII Card Codes

ASCII Character	Octal Code	Card Punches	ASCII Character	Octal Code	Card Punches
NULL	00	12-0-9-8-1	@	100	8-4
CTRL-A	01	12-9-1	A	101	12-1
CTRL-B	02	12-9-2	B	102	12-2
CTRL-C	03	12-9-3	C	103	12-3
CTRL-D	04	9-7	D	104	12-4
CTRL-E	05	0-9-8-5	E	105	12-5
CTRL-F	06	0-9-8-6	F	106	12-6
CTRL-G	07	0-9-8-7	G	107	12-7
CTRL-H	10	11-9-6	H	110	12-8
TAB	11	12-9-5	I	111	12-9
LF	12	0-9-5	J	112	11-1
VT	13	12-9-8-3	K	113	11-2
FF	14	12-9-8-4	L	114	11-3
CR	15	12-9-8-5	M	115	11-4
CTRL-N	16	12-9-8-6	N	116	11-5
CTRL-O	17	12-9-8-7	O	117	11-6
CTRL-P	20	12-11-9-8-1	P	120	11-7
CTRL-Q	21	11-9-1	Q	121	11-8
CTRL-R	22	11-9-2	R	122	11-9
CTRL-S	23	11-9-3	S	123	0-2
CTRL-T	24	9-8-4	T	124	0-3
CTRL-U	25	9-8-5	U	125	0-4
CTRL-V	26	9-2	V	126	0-5
CTRL-W	27	0-9-6	W	127	0-6
CTRL-X	30	11-9-8	X	130	0-7
CTRL-Y	31	11-9-8-1	Y	131	0-8
CTRL-Z	32	9-8-7	Z	132	0-9
ESCAPE	33	0-9-7	[	133	12-8-2
CTRL-\	34	11-9-8-4	\	134	0-8-2
CTRL-]	35	11-9-8-5	]	135	11-8-2
CTRL-^	36	11-9-8-6	^	136	11-8-7
CTRL-`	37	11-9-8-7	`	137	0-8-5
SPACE	40		,	140	8-1

NOTE: The ASCII character ESCAPE (octal 33) is also CTRL-[ on a terminal.

Table B-1 (Cont.)

## ASCII Card Codes

ASCII Character	Octal Code	Card Punches	ASCII Character	Octal Code	Card Punches
!	41	12-8-7	a	141	12-0-1
'	42	8-7	b	142	12-0-2
#	43	8-3	c	143	12-0-3
\$	44	11-8-3	d	144	12-0-4
%	45	0-8-4	e	145	12-0-5
&	46	12	f	146	12-0-6
'	47	8-5	g	147	12-0-7
(	50	12-8-5	h	150	12-0-8
)	51	11-8-5	i	151	12-0-9
*	52	11-8-4	j	152	12-11-1
+	53	12-8-6	k	153	12-11-2
,	54	0-8-3	l	154	12-11-3
-	55	11	m	155	12-11-4
.	56	12-8-3	n	156	12-11-5
/	57	0-1	o	157	12-11-6
o	60	0	p	160	12-11-7
1	61	1	q	161	12-11-8
2	62	2	r	162	12-11-9
3	63	3	s	163	11-0-2
4	64	4	t	164	11-0-3
5	65	5	u	165	11-0-4
6	66	6	v	166	11-0-5
7	67	7	w	167	11-0-6
8	70	8	x	170	11-0-7
9	71	9	y	171	11-0-8
:	72	8-2	z	172	11-0-9
;	73	11-8-6		173	12-0
<	74	12-8-4		174	12-11
=	75	8-6		175	11-0
>	76	0-8-6		176	11-0-1
?	77	0-8-7	DEL	177	12-9-7

NOTE: The ASCII characters and (octal 175 and 176) are treated by the monitor as ALTmode which is often considered to be the same as ESCAPE.

TABLE B-2

## DEC-029 Card Codes

Character	Octal Code	Card Punches	Character	Octal Code	Card Punches
SPACE	40		@	100	8-4
!	41	11-8-2	A	101	12-1
'	42	8-7	B	102	12-2
#	43	8-3	C	103	12-3
\$	44	11-8-3	D	104	12-4
%	45	0-8-4	E	105	12-5
&	46	12	F	106	12-6
,	47	8-5	G	107	12-7
(	50	12-8-5	H	110	12-8
)	51	11-8-5	I	111	12-9
*	52	11-8-4	J	112	11-1
+	53	12-8-6	K	113	11-2
,	54	0-8-3	L	114	11-3
-	55	11	M	115	11-4
.	56	12-8-3	N	116	11-5
/	57	0-1	O	117	11-6
0	60	0	P	120	11-7
1	61	1	Q	121	11-8
2	62	2	R	122	11-9
3	63	3	S	123	0-2
4	64	4	T	124	0-3
5	65	5	U	125	0-4
6	66	6	V	126	0-5
7	67	7	W	127	0-6
8	70	8	X	130	0-7
9	71	9	Y	131	0-8
:	72	8-2	Z	132	0-9
:	73	11-8-6	[	133	12-8-2
<	74	12-8-4	\	134	11-8-7
=	75	8-6	]	135	0-8-2
>	76	0-8-6	^	136	12-8-7
?	77	0-8-7	_	137	0-8-5

NOTE: Octal codes 0-37 and 140-177 are the same as in ASCII.

Table B-3

## DEC-026 Card Codes

Character	Octal Code	Card Punches	Character	Octal Code	Card Punches
SPACE	40		@	100	8-4
!	41	12-8-7	A	101	12-1
'	42	0-8-5	B	102	12-2
#	43	0-8-6	C	103	12-3
\$	44	11-8-3	D	104	12-4
%	45	0-8-7	E	105	12-5
&	46	11-8-7	F	106	12-6
,	47	8-6	G	107	12-7
(	50	0-8-4	H	110	12-8
)	51	12-8-4	I	111	12-9
*	52	11-8-4	J	112	11-1
+	53	12	K	113	11-2
,	54	0-8-3	L	114	11-3
-	55	11	M	115	11-4
.	56	12-8-3	N	116	11-5
/	57	0-1	O	117	11-6
0	60	0	P	120	11-7
1	61	1	Q	121	11-8
2	62	2	R	122	11-9
3	63	3	S	123	0-2
4	64	4	T	124	0-3
5	65	5	U	125	0-4
6	66	6	V	126	0-5
7	67	7	W	127	0-6
8	70	8	X	130	0-7
9	71	9	Y	131	0-8
:	72	11-8-2/11-0	Z	132	0-9
;	73	0-8-2	[	133	11-8-5
<	74	12-8-6	\	134	8-7
=	75	8-3	]	135	12-8-5
>	76	11-8-6	^	136	8-5
?	77	12-8-2/12-0	_	137	8-2

NOTE: Octal codes 0-37 and 140-177 are the same as in ASCII.



## APPENDIX C TEMPORARY FILES

The temporary files in Table C-1 are used by various programs in the DECsystem-10 computing system. These files are in the following form:

nnn xxx.TMP

where nnn is the user's job number in decimal, with leading zeroes to make three digits, and xxx specifies the use of the file.

TABLE C-1  
Temporary Files

Name	Meaning
nnn ALG.TMP	Read by ALGOL and contains one line for each program to be compiled. It may also contain the command NAME! which causes ALGOL to transfer control to the named program.
nnn AS1.TMP nnn AS2.TMP nnn AS3.TMP	Written, read, and deleted by COBOL and contains input to the COBOL assembler.
nnn BLI.TMP	Read by BLISS and contains one line for each program to be compiled.
nnn COB.TMP	Read by COBOL and contains one line for each program to be compiled. It may also contain the command NAME! which causes COBOL to transfer control to the named program.
nnn CPY.TMP	Written, read, and deleted by COBOL and contains copies of source files with library routines inserted.
nnn CRE.TMP	Read by CREF and contains commands for each file which has produced a CREF listing on the disk. COMPIL also reads this file each time a new CREF listing is generated to prevent multiple requests for the same file and to prevent discarding other requests that may not yet have been listed.

Table C-1 (Cont.)

## Temporary Files

Name	Meaning
nnn DAE.TMP	Written by DAEMON to be read by DUMP.
nnn DMP.TMP	Read by DUMP as an input command file.
nnn EDS.TMP	Used by COMPIL to store the arguments of the most recent EDIT, CREATE, TECO, or MAKE command.
nnn EDT.TMP	<p>Written by COMPIL and read by LINED or TECO. It contains a command for each EDIT, CREATE, TECO, or MAKE command. For the MAKE or CREATE commands, it contains the command</p> <p style="text-align: center;">S file.ext [p,p] \$</p> <p>For TECO or EDIT commands, it contains the command</p> <p style="text-align: center;">S file.ext [p,p]</p>
nnn ERA.TMP	Written, read, and deleted by COBOL and is the error file.
nnn FAI.TMP	Read by FAIL and contains one line for each program to be compiled.
nnn FOR.TMP	Read by FORTRAN and contains one line for each program to be compiled. It may also contain the command NAME! which causes FORTRAN to transfer control to the named program.
nnn GEN.TMP	Written, read, and deleted by COBOL and contains the output of syntax processing.
nnn KJO.TMP	Read by KJOB as an input command file.
nnn LGO.TMP	Read by LOGOUT as an input command file.
nnn LHC.TMP	Created and read by LINK-10 and contains the overflow of the user's high segment. The file is used to produce core images or saved files.
nnn LIN.TMP	Created by LINED and contains output file until the rename process.
nnn LIT.TMP	Written, read, and deleted by COBOL and contains copy of the literal pool.

Table C-1 (Cont.)

## Temporary Files

Name	Meaning
nnn LLC.TMP	Created and read by LINK-10 and contains the overflow of the user's low segment. This file is used to produce core images or saved files.
nnn LLS.TMP	Created and read by LINK-10 and contains the overflow of the user's symbol file. This file is used to produce core images or saved files.
nnn LNK.TMP	Read by LINK-10 and contains commands necessary for loading.
nnn LOA.TMP	Read by LOADER and contains commands necessary for loading.
nnn MAC.TMP	Read by MACRO and contains one line for each program to be assembled. It may also contain the command NAME! which causes MACRO to transfer control to the named program.
nnn P11.TMP	Read by MACX11 (the PDP-11 assembler for the PDP-10) and contains one line for each program to be assembled.
nnn PLS.TMP	Read by PLEASE as an input command file.
nnn PIP.TMP	Read by PIP and contains commands to implement the COMPIL-class commands that run PIP.
nnn QUE.TMP	Read by QUEUE as an input command file.
nnn RNO.TMP	Read by RUNOFF and contains commands for each file which has produced a RUNOFF listing on the disk.
nnn S01.TMP	Written, read and deleted by COBOL and contains the intermediate sorted results of the data.
nnn SVC.TMP	Used by COMPIL to store the arguments of the most recent COMPILE, LOAD, EXECUTE, or DEBUG command.
nnn SNO.TMP	Read by SNOBOL and contains one line for each program to be compiled.
nnn TEC.TMP	Created by TECO and contains output file until the rename process.

Table C-1 (Cont.)

Temporary Files

<b>Name</b>	<b>Meaning</b>
nnn TMP.TMP	Created by LINED during the rename process.
nnn XFO.TMP	Created by FILEX as a result of the Q switch on the output side.
nnn XFR.TMP	Created by FILEX as a result of the Q switch on the input side.

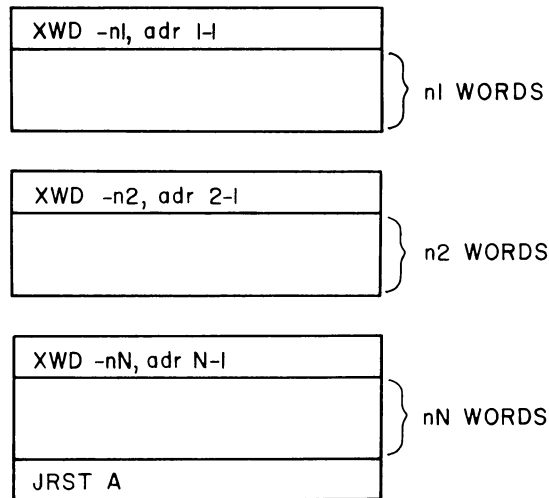
## APPENDIX D

### SAVE AND SSAVE COMMANDS

Before writing SAVed or LOW files in response to SAVE and SSAVE commands (refer to the individual command descriptions in Chapter 2), the monitor compresses the user's core image by eliminating consecutive blocks of zeroes. This technique is known as zero-compression and is used to save space on file media. Low segment files are zero-compressed on devices DTA, MTA, and DSK, but high segment files are not because the high segment can be shared at the time of the command.

SAVed files are ordinary binary files and can be copied using the /B switch in PIP. Files with the LOW or SAV extension may be read in dump mode, but must be reexpanded before being run. The monitor expands the file after input on a RUN, R, or GET command. The FILEX program may be used to expand the file for other purposes.

The data format of a zero-compressed SAVed file consists of a series of IOWDs and data block pairs and is terminated by a JRST A where A is the program starting address as specified by the contents of .JBSA. The format is as follows:

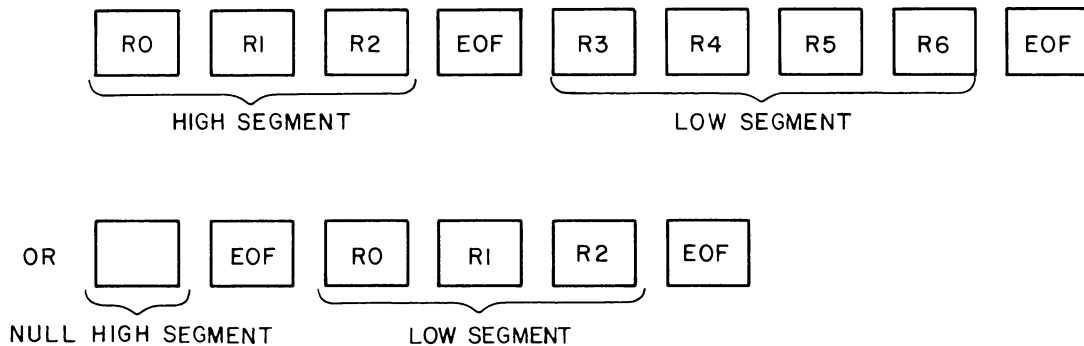


Each IOWD describes the length of the following data block and the original location of the data in core. The LH of the IOWD can be positive in which case the number of words is taken as the number of words greater than 128K.

SAVed files are read into the user's core area starting at location .JBSAV and then are expanded to occupy the original relative locations. If the first word read is not an IOWD and is positive, an old-format, noncompressed saved file is assumed and no expansion is performed.

A SAVE command issued to a magnetic tape writes

- a. a high segment (possibly null)
- b. an EOF
- c. a low segment (possibly null)
- d. an EOF.



The monitor does not determine the file size of a low segment on a GET from magnetic tape; therefore, a user must always specify a core argument or have enough core assigned to his job for the file.

To save file space, only the high segment up through the highest nonzero location (relative to high segment origin) loaded, as specified in the LH of .JBHRL, will be written by the SAVE command. If LH is zero (high segment created by CORE or REMAP UUO) or DDT is present, the entire high segment will be written.

The LOADER indicates to the SAVE command how much data was loaded above the job data area in the low segment by setting the LH of .JBCOR to the highest location in the low segment that was not explicitly loaded with data (either zero or nonzero). Most programs are written so that only the high segment contains nonzero data. In this case, SAVE and SSAVE write only the high segments. This also saves file space and I/O time with the GET command.

A number of locations in the job data area need to be initialized on a GET, although there is no other data in the low segment. The SAVE command copies these locations into the first 10(octal) locations of the high segment, provided it is not sharable. The locations are referred to as the vestigial job data area (refer to DECsystem-10 Monitor Calls, Chapter 1). Therefore, the LOADER will load high segment programs starting at location 400010.

To prevent user confusion, SAVE and SSAVE delete a previous file with the extension .SHR or .HGH; therefore, SAVE deletes a file with the extension .SHR, and SSAVE deletes a file with the extension .HGH. SAVE and SSAVE commands also delete files with the extension .LOW, if the high segment was the only segment written.

The regular access rights of the saved file indicate whether a user can perform a GET,R, or RUN command. These commands assume that the user wants to execute (but not modify) the high segment, independent of the access rights of the file used to initialize the segment. The monitor always enables the hardware user-mode write protect to prevent the user program from storing into the segment inadvertently.

To debug a reentrant system program, the user should make a private, nonsharable copy, rather than modify the shared version and possibly cause harm to other users. To make a private, nonsharable copy, the following commands are used:

GET prog	
SAVE	Writes a file in the user directory as nonsharable. The high segment in the user's addressing space remains sharable.
GET	Overlays the sharable program with the nonsharable one from the user's directory. Now the user can make patches while other users share the version in the library.

If the user is debugging a sharable program in his UFD with the D command or the DDT program, it is recommended that the program be nonsharable instead of sharable. The reason for this is that the user may wish to modify the high segment during the debugging phase and later reinitialize the original unmodified high segment from the file with a GET command. However, since the high segment is sharable, the monitor will not do I/O into it, will not reinitialize it from the disk file, and the user will receive the modified high segment instead.

NOTE

DDT modifies the high segment when it inserts breakpoints.

The following examples are the incorrect and correct methods of debugging a sharable program. After the debugging phase is completed, the SSAVE command should be used to save the program.

Example 1: Incorrect Method

.DEBUG prog	
EXECUTION	
^C	
.SSAVE	;SAVE should be used in debugging
.GET	
JOB SETUP	
.E 40010	
400010/777777 777777 .D 0 0	
.E	
400010/ 0 0	
.GET	
JOB SETUP	
.E 40010	
400010/ 0 0	;not the original 777777 777777

## Example 2: Correct Method

```
.DEBUG prog
EXECUTION
^C
.SAVE
.GET
JOB SETUP
.E 400010
400010/777777 777777.D 0 0
.E
400010/ 0 0
.GET
JOB SETUP
.E 400010
400010/777777 777777 ;the original file
```

Note that there are applications for a sharable data segment when the modified version of the sharable segment is wanted rather than the original segment as initialized from the file. The SSAVE command is then used.

A SAVE of a one-segment program and a SSAVE of a two-segment program of the same name can coexist in the same directory, and the monitor keeps the two versions separate. This allows for a common library, of reentrant and non-reentrant versions of the same system programs to service both the PDP-6 and the DECsystem-10. A sharable program may be superseded into the directory by the SSAVE command. The monitor clears the high segment in its table of sharable segments in use but does not remove the segment from the addressing space of users currently using it. Only the users doing a GET,R, or RUN command or a RUN or GETSEG UUC have the new sharable version.

When the SAVE or SSAVE command is used to save a sharable program with only a high file, the monitor does not modify the vestigial job data area. This prohibits unauthorized users from modifying the first 10 locations of a shared segment by executing a SAVE or SSAVE command. This restriction does not exist if a low file is also written, because the GET command reads the low file after the high file, so that the real job data area locations are set from the low file. To change the version number of a sharable two-segment program with only a high file, the following commands are used.

```
GET prog
SAVE
GET
D nnn mmm 137
SSAVE
```

The SAVE command makes the program non-sharable so that the vestigial job data area can be modified by the SSAVE.



## APPENDIX E

### SWITCHES

A switch is one or more characters preceded by a slash. Its appearance within a command line causes a modification to be made to the meaning of the command line. There are two classes of switches; namely, permanent switches and temporary switches.

A permanent switch applies to all filenames appearing after it within the command line.

```
Keyword file1 /switch file2 file3
```

In this example, the permanent switch pertains to files 2 and 3. If a filename precedes the permanent switch, that filename and switch are separated by one or more commas and/or spaces.

A temporary switch applies only to that filename that directly precedes the switch.

```
keyword file1/switch file2 file3
```

No intervening spaces or commas are allowed between the temporary switch and its associated filename.

Switches may be abbreviated; the letters which must be declared are those that will uniquely specify the desired switch. For instance, if a command can use one of the following switches:

1. /ABCD
2. /ABC
3. /ABBDE
4. /ACBDE

In order to uniquely specify the switches, the user could abbreviate the switches to:

1. /ABCD
2. /ABC
3. /ABB
4. /AC

Some examples of permanent and temporary switches are:

```
. COMPILE PROG,TEST,MAC,MANAGE/COBOL
```

Compile three files: namely, PROG (with a null extension) with the default processor FORTRAN, TEST.MAC with MACRO, and MANAGE with COBOL. The /COBOL switch is a temporary switch applying only to the filename directly preceding it.)

. COMPILE /COBOL,PROG,TEST,MAC,MANAGE

Compile three files; namely, PROG and MANAGE with COBOL and TEST,MAC with MACRO. A filename written with a recognizable processor extension (i.e.,MAC) will always be processed by the processor specified by the extension. A switch, permanent or temporary, cannot override a recognizable processor extension. The /COBOL switch is a permanent switch applying to those filenames appearing after it.

. CPUNCH ABC,MAC/PUNCH:ASCII TEST

Punch ABC,MAC in ASCII mode; and punch TEST in the data mode of the file. The /PUNCH switch is a temporary switch within this example.

. CPUNCH /PUNCH:BINAR,ABC,MAC,TEST,FOO.

Punch ABC,MAC, TEST and FOO in binary card code. The /PUNCH switch is a permanent switch within this example.

The table appearing on the following pages contains a list of the switches described in this manual. For each switch there appears a list of commands and/or programs which can utilize the specified switch and the switch's meaning within the associated command line.

Switch	Applicable Commands/ Programs	Meaning
/A	FAILSA	Advance magnetic tape one physical unit.
	FILCOM	Compare files in ASCII mode. This switch is used to force a source compare on two ASCII files.
	FILEX	File format is ASCII; meaningful only for PDP-11 and PDP-15 tapes.
	GLOB	Output all global symbols.
/ACCESS:n	DIRECT	Update the access date to the current date for any file of n blocks or less accessed (i.e., listed) by the DIRECT program. Since some installations delete files that have not been recently accessed, this switch allows the user to prevent such deletion by updating the date. n is interpreted as a decimal number and refers to the number of blocks actually written in the file unless the /ALLOC switch is also used. If /ACCESS is omitted, the date is not changed. If /ACCESS is specified but :n is omitted, n = 5 is assumed.

Switch	Applicable Commands/ Programs	Meaning
/AFTER:t	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Process the request after the specified time. Time is specified in the form described on page 1-12. The resulting AFTER time must be less than the DEADLINE time. If either the switch or the value is omitted, no AFTER constraints are assumed.
/ALGOL	COMPILE DEBUG EXECUTE LOAD	Compile the file with ALGOL. Assumed for files with the ALG extension.
/ALLOC	DIRECT	List the allocated length of the file instead of the written length. Space on a structure is sometimes allocated in units of more than one block for efficiency. Therefore, the number of blocks allocated to a file may be greater than the number of blocks actually written. The allocated length is used by LOGOUT in checking quotas. The total allocated length of all files is the same as the length output by the QUOLST program under the USED column. Complement of /WRITTEN. (Disk and magnetic tape only.)
/B	FAILSA FUDGE2	Backspace the magnetic tape one physical file.
	FILCOM	Compare blank lines. Without this switch blank lines are ignored.
	FILEX	Process a binary file; overrides the default extension. Files read from a PDP-11 format tape with this switch contain four 8-bit bytes in each 36-bit word.
/BEFORE:t	CPUNCH DIRECT PLOT PRINT QUEUE TPUNCH FORTRAN	List only the files with a creation date before time t, where t is in the form dd-mm-yy hh:mm. If the switch, or the value of the switch, is omitted, no BEFORE constraints are assumed.

Switch	Applicable Commands/ Programs	Meaning
/BEGIN:n	CPUNCH PLOT PRINT QUEUE TPUNCH PUNCH	Start the output on the nth page, card, or foot. If this switch is omitted, output begins on the first unit.
/BIN	COMPILE DEBUG EXECUTE LOAD	Generate a binary file for each file compiled. The filename of the binary file follows the standard convention for determining the filename of the output file. The extension is .REL. This is the default action, whenever a COMPILE, DEBUG, EXECUTE or LOAD command is given.
/BLISS	COMPILE DEBUG EXECUTE LOAD	Compile this file with BLISS-10. Assumed for files with the .B10 or BLI extension.
/BLOCKS	DIRECT	Output the length of the file in blocks instead of words. Complement of /WORDS. /BLOCKS is the default setting.
/C	DTCOPY  FAILSA  FILCOM  FILEX	Copy all blocks from the input DECTape to the output DECTape.  Cause FAILSA to continue; particulars are detailed within the description of the FAILSA program.  Ignore comments (all text on a line following a semicolon) and spacing (spaces and tabs).  Use a compressed file format; save file format. This format is assumed for files with the extensions .SAV, .LOW, or .SVE. The default output extension is .SAV unless the input extension is .LOW or .SVE, in which case the extension remains unchanged.
/CARDS:n	QUEUE SUBMIT CPUNCH	Use n (decimal) as the maximum number of cards that can be punched by this job. If the switch is omitted, no cards are punched. If the switch is given with no value for n specified, 2000 cards are assumed.

Switch	Applicable Commands/ Programs	Meaning
/CHARGE:a	QUEUE	Charge the run to the specified account. (Not yet implemented.)
/CHECK	DISMOUNT MOUNT	Check and list pending requests.
/CHECKSUM	DIRECT	Compute and print an 18-bit checksum for each file. Complement of /NOCHECKSUM. (Disk and magnetic tape only.)
/COBOL	COMPILE DEBUG EXECUTE LOAD	Compile the file with COBOL. Assumed for files with the .CBL extension.
/COMPILE	COMPILE DEBUG EXECUTE LOAD	Force a compilation of this file even if a binary file exists with a newer date and time than the source file. This switch is used to obtain an extra compilation (e.g., in order to obtain a listing of the compilation) because normally compilation is not performed if the binary file is newer than the source file.
/COPIES:n	CPUNCH PLOT PRINT QUEUE TPUNCH	Repeat the output the specified number of times: n must be less than 64. If more than 63 copies are needed, two separate requests must be made. If this switch or its value is not specified, the default number of copies obtained is one.
/CORE:n	QUEUE SUBMIT	Use n (in decimal K) as the maximum of core memory that the job can use. If the switch is omitted, the maximum of 25K is assumed; if the switch is specified but the value of n is omitted, a maximum of 40K is assumed.
/CREATE	CPUNCH PLOT PRINT TPUNCH QUEUE SUBMIT  SETSRC	Make a new entry into the card, plotter line printer, paper tape, or specified output/input queue. This switch is the default for the queue-operation switches.  Allow new files to be created on the file structure.

Switch	Applicable Commands/ Programs	Meaning
/CREF	COMPILE DEBUG EXECUTE LOAD	Produce a cross-reference listing file on the disk for each file compiled for later processing by the CREF program. The filename for the listing file follows the standard conventions for determining the name of the output file. The extension is .CRF.
/D	FAILSA  FILEX	Transfer control to DDT if it is loaded.  Use dump file format. This format is assumed for files with the .DMP extension.
/DDT	DEBUG	Load DDT regardless of the extension of the first file in the command string. This is a permanent switch and applies to all subsequent files.
/DEADLINE:tt	CPUNCH PLOT PRINT TPUNCH QUEUE SUBMIT	Process the request before the specified time. Time is specified in the form described on page 1-12. The resulting DEADLINE time must be greater than the AFTER time. If the switch or its value is omitted no DEADLINE constraints are assumed.
/DEFER	QUEUE	Make a new entry in the specified queue, but the request is deferred until LOGOUT. (Not yet implemented.)
/DENSITY:n	DIRECT	Use the specified density when reading a magnetic tape. n is either 200, 556, or 800 bpi. The default is installation dependent and is modified by the SET DENSITY command.
/DELETE	QUEUE	Rename the file and delete name from the directory immediately. Remove the file from the logged out quota, and delete it after spooling.
/DEPEND:n	QUEUE SUBMIT	Specify the initial value of the dependency count in decimal. When used with /MODIFY, this switch changes the dependency count of another job. If n is a signed number, that number is added or subtracted from the count of the current job. If n is not a signed number, the dependent job's dependency count is equal to n. If this switch is omitted, no dependency count is assumed.

Switch	Applicable Commands/ Programs	Meaning
/DETAIL	DIRECT	Print all available information about a file except for zero values. The protection and data mode are also listed, even if they have a value of zero. The author is not listed if he is the same as the owner of the directory. Numbers followed by a period are decimal numbers. All others are interpreted as octal. (Disk and magnetic tape only.)
/DISPOSE: DELETE	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Delete the file after it has been punched, plotted, printed, spooled, or processed.
/DISPOSE: PRESERVE	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Do not delete the file after it has been processed.
/DISPOSE: RENAME	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Rename the file and delete current name from the directory immediately (for QUEUE) or after it has been processed for CPUNCH, PLOT, PRINT, SUBMIT, or TPUNCH). Remove the file from the logged-out quota, and delete it after spooling (for QUEUE).
/E	FAILSA	Set the creation date and time word so that old files can be eliminated by not being transferred in either direction.  The format is as follows: */E mm/dd/yy.ttt<CR>
	FILEX	Use expanded core image file format (used by FILDDT). This format is assumed for files with the extension of .XPN. The default output extension is .XPN.
	GLOB	List only erroneous (multiply-defined and undefined) symbols.

Switch	Applicable Commands/ Programs	Meaning
/EOTS	DIRECT	Stop at the logical end of the tape (two consecutive tape marks) when reading a magnetic tape. Complement of /NOEOTS. /EOTS is the default condition.
/F	FAILSA	Set the access date word. This switch is used to transfer, in either direction, only recently accessed files or to save on tape and then delete from disk files not recently accessed (/K before /S or /U).
	FILEX	Use PDP-15 DECTape format.
	GLOB	List non-relocatable (fixed) symbols only.
	CPUNCH PLOT PRINT PUNCH QUEUE SUBMIT TPUNCH	List the entries in the specified queue. but do not update the queue.
/FAST	DIRECT	List the short form of the directory (i.e., filename, extension, structure name, and directory name). Complement of /NORMAL and /SLOW.
/FEET:n	QUEUE SUBMIT	Use n (in decimal) as the maximum number of feet of paper tape that the job can punch. If the switch is omitted, no paper tape is punched. If the value is omitted, the default is $10*B+20$ , where B is the number of blocks in the request.
/FILES:n	DIRECT	Stop after n tape marks (files) when reading a magnetic tape. If /FILES is specified, but :n is omitted, n = 1 is assumed. Note that the logical EOT will also stop unless /NOEOTS is specified.
/FILES: ASCII	PRINT QUEUE TPUNCH	Interpret as ASCII text. This is assumed for all files with extensions other than .DAT.
/FILES: COBOL	PRINT QUEUE	Interpret the file format as COBOL SIXBIT text.



Switch	Applicable Commands/ Programs	Meaning
/FILES: ELEVEN	QUEUE TPUNCH	Specify that the file form is to be interpreted as four 8-bit bytes in each 36-bit word.
/FILES: FORTRAN	PRINT QUEUE	Interpret the file format as FORTRAN ASCII (obeys FORTRAN control characters). This is assumed for files with an extension of .DAT.
/FOROTS	EXECUTE DEBUG LOAD	Load the file with FOROTS (the new FORTRAN object-time system).
/FORSE	EXECUTE DEBUG LOAD	Load the file with FORSE (the old FORTRAN object-time system).
/FORMS:a	PLOT PRINT QUEUE	Place the output on the specified forms. The argument to the switch must be six alphanumeric characters in length. Normal forms (14 x 11) are used if this switch is omitted.
/FUDGE	COMPILE DEBUG EXECUTE LOAD	Create a disk file containing the names of the .REL files produced as a result of the command string. Arguments to this switch are:  /FUDGE:dev:file.ext[proj.prog]
/F10	COMPILE DEBUG EXECUTE LOAD	Use the FORTRAN-10 compiler when compiling the associated FORTRAN file. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.
/F40	COMPILE DEBUG EXECUTE LOAD	Use the F40 compiler when compiling the associated FORTRAN file. This is the current default action. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.
/G	DTCOPY	Do not restart the program after a parity error. Output an error message and continue the program.

Switch	Applicable Commands/ Programs	Meaning
/H	FAILSA	<p>Enable the user to save and restore files from a user's area other than his own. This switch sets the source project-programmer number to the specified number of the form:</p> <p style="text-align: center;">*/G mmm.nn&lt;CR&gt;</p> <p>The new value is retained until the next /G switch is issued.</p>
	FILEX	<p>Ignore read errors on the input device and continue FILEX. Checks the always-bad-checksum bit in the 5-series monitor, so this switch is not needed for files with .RPABC ON (e.g., CRASH.SAV).</p>
/HEADER: 0 or 1	DTCCOPY DIRECT FAILSA FILCOM FILEX GLOB MOUNT	<p>Print messages which will be of help to the user.</p>
/HELP	QUEUE	<p>If n equals 1 (default), do not output headers. If 0, output block headers at beginning of file.</p>
/HELP: S	DIRECT QUEUE	<p>Print a message giving the general format of the command string and explain the dialogue that is entered if the user needs additional help.</p>
/I	FAILSA	<p>List all switches without their explanations. An asterisk (*) prefixes those switches which have a single-letter abbreviation.</p>
	FILEX	<p>Set the magnetic tape density to the installation standard.</p>
/J	FAILSA	<p>Process an image mode file: meaningful only for PDP-11 and PDP-15 tapes.</p>
		<p>Look for the next trailer record (which marks the end of the save set) and point to the beginning of the next save set on the tape or to the logical end of the tape, if there is no save set.</p>

Switch	Applicable Commands/ Programs	Meaning
/K	FAILSA	Delete not-recently accessed files from disk and copy them to a magnetic tape. (Used in conjunction with /S or /U.)
	FUDGE2	Advance a magnetic tape one file.
/KA10 /KI10	COMPILE DEBUG EXECUTE FORTRAN LOAD	Designate the machine on which the program will execute once it has been loaded.
/KILL	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Remove the specified entry from the specified queue. The /KILL switch can be used for deleting a previously submitted request, as long as the request has not been started.
/nL	FILCOM	Specify the lower limit for a partial binary compare (n is an octal number). This switch when used with /nU, allows a binary file to be compared only within specified limits.
/L	DTCOPY	Load the bootstrap loader into a core buffer. DTCOPY expects the loader to be on logical device PTR in the file named RSLDR.REL. Note that DTCOPY must be SAVED if the loader is to be preserved with the DTCOPY core image.
	FAILSA	Type on the user's terminal, a directory of all the user's files on the tape. FAILSAFE checks the current user's project-programmer number and uses it to find the correct area of the tape. Only the filenames and extensions are typed.
	FILEX	Type the directory of an input DECTape file on the terminal, or list the directory of the output DECTape at the end (i.e., after the output).
	GLOB	Scan programs only if they contain globals previously defined and not yet satisfied. (Library search mode.)
	SETSRC	Set the job's library directory to the UFD [proj.prog] and add it to the user's DSK specification.
/LIB: [proj.prog]		

Switch	Applicable Commands/ Programs	Meaning
/LIBRARY	DEBUG EXECUTE LOAD	Load the files in library search mode. The system libraries are always searched.
/LIMIT:n	CPUNCH PLOT PRINT QUEUE TPUNCH	Limit the output to the specified number of cards, pages, feet or minutes.
/LINK	COMPILE DEBUG EXECUTE LOAD	Load the file with the LINK-10 linking loader. If used, this switch should be placed before any file specifications since the COMPIL program may have to generate load-control switches. Complement of /LOADER.
/LIST	COMPILE DEBUG LOAD EXECUTE	Generate a disk listing file, for each file compiled. The filename for the listing file follows the standard conventions for determining the name of the output file. The extension is .LST. These files can later be listed with the LIST command.
	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	List the specified entries in the specified queue. If the switch is omitted, all entries for all jobs of all users are listed.
/LMAP	DEBUG EXECUTE LOAD	Produce a loader map during the loading process (same action as MAP) containing the local symbols.
/LOADER	DEBUG EXECUTE LOAD	Load the file with the LOADER program. This is the current default action. Complement of /LINK.
/LOG	PRINT QUEUE	Define the file that the spoolers will use to record their output. The default is jobname .LOG.

Switch	Applicable Commands/ Programs	Meaning
/M	FAILSA	Initiate multiple saves. Saves are taken every SLPMIN minute, where SLPMIN is an assembly parameter initially set to 60 (decimal) minutes.
	FILEX	Use MIT project PDP-6/10 DECtape format.
	GLOB	Turn off library search mode scanning resulting from a /L switch.
/MACRO	DEBUG EXECUTE LOAD	Assemble the file with MACRO. Assumed for files with extensions of .MAC.
/MACX11	COMPILE DEBUG EXECUTE LOAD	Assemble the file with MACX11. Assumed for files with the .P11 extension.
/MANTIS	COMPILE DEBUG EXECUTE LOAD	Compile the file with MANTIS debugging information. This switch affects FORTRAN programs only.
/MAP	DEBUG EXECUTE LOAD	Produce a loader map during the loading process. This switch is an exception to the permanent switch rule in that it causes only one map to be produced even though it may appear as a permanent switch.
/MARKS	DIRECT	Indicate each tape mark, including the final tape mark, and UFD when reading a magnetic tape. Complement of /NOMARKS.
/MODIFY	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Alter the specified parameters in the job. This switch requires that the user have access rights to the job.
/MULTI	MOUNT	Permit multi-access (disk only). Complement of /SINGLE. The function of this switch is the default condition.

Switch	Applicable Commands/ Programs	Meaning
/N	DTCOPY	Suppress the directory listing.
	FAILSA	Do not check the creation dates of tape and disk files of same names when restoring from tape. This switch remains in effect until the line of input is terminated by a carriage return-line feed.
	GLOB	List only the symbols which are never referenced.
/NEW	COMPILE DEBUG EXECUTE LOAD	Run the appropriate language translator from the experimental system library (device NEW:) area [1.5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:.
	CPUNCH PLOT PRINT QUEUE TPUNCH	Accept the request even if the file does not yet exist. An appropriate error message is given if the file does not exist by the time the request is processed by the spooler. For the QUEUE and SUBMIT commands, this is the default setting for the log file of the Batch input queue. When using SUBMIT, the user could place this switch within his control file, allowing him to submit his job and then create the control file.
	SETSRC	Add the directory [1.5] to the user's SYS specification. This means that when the system directory is searched, the directory [1.5] will be searched before the directory [1.4].
/NOBIN	COMPILE DEBUG EXECUTE LOAD	Do not generate binary files. Unless this switch is given, binary files are generated. This switch, when combined with /LIST or /CREF, is useful when compiling programs solely for the purpose of generating listings.
/NOCHECKSUM	DIRECT	Do not compute and print the checksum. Complement of /DETAIL. This is the default condition.
/NOCOMPILE	COMPILE DEBUG EXECUTE LOAD	Complement the /COMPILE switch by not forcing a compilation on a source file whose date is not as recent as the date on the binary file. Note that this switch is not the same as the /REL switch, which turns off all compilation, even if the source file is newer than the REL file. /NOCOMPILE is the default condition.

Switch	Applicable Commands/ Programs	Meaning
/NOCREATE	SETSRC	Do not allow new files to be created on the file structure when DSK is specified, but allow files to be updated. Files can be created on the file structure if the user specifies the file structure name explicitly.
/NODETAIL	DIRECT	Do not list the words in the LOOKUP block. Complement of /DETAIL. This is the default condition.
/NOEOTS	DIRECT	Do not stop at the logical end of the tape when reading a magnetic tape. Complement of /EOTS.
/NOLIB	SETSRC	Remove the library directory from the user's DSK specification.
/NOLIST	COMPILE DEBUG EXECUTE LOAD	Do not generate listing files. This is the default condition.
/NOMANTIS	COMPILE DEBUG EXECUTE LOAD	Compile the program without the MANTIS debugging information. This switch affects FORTRAN files only.
/NOMARKS	DIRECT	Do not indicate each tape mark and UFD when reading a magnetic tape. Complement of /MARKS. This is the default condition.
/NONEW	SETSRC	Remove the directory [1,5] from the user's SYS specification.
/NORMAL	DIRECT	Output the normal directory listing. This listing includes the filename, extension, length in blocks written, protection, creation date, structure name, non-zero version numbers, and directory name. Complement of /FAST and /SLOW. This is the default condition. This switch is also used to override a /FAST or /SLOW in the user's option file.

Switch	Applicable Commands/ Programs	Meaning
/NOREWINDS	DIRECT	Do not rewind the tape before and after reading a magnetic tape. Complement of /REWINDS.
/NOSCAN	SETSRC	Cancel the scan switch (/SCAN) for the directory path.
/NOSEARCH	DEBUG EXECUTE LOAD	Load all routines of the file whether or not the routines are referenced. Since this is the default condition, this switch is used only to turn off library search mode (/LIBRARY). This switch is not the equivalent of the /P switch of the LOADER, which does not search any libraries. The /NOSEARCH default searches the system libraries.
/NOSORT	DIRECT	Do not produce a file suitable for sorting. Complement of /SORT. This is the default condition.
/NOSUMMARY	DIRECT	Do not use summary mode; i.e., output more than just the summary line. Complement of /SUMMARY. This is the default condition.
/NOSYS	SETSRC	Remove the SYS specification from the user's DSK specification.
/NOTE:a	PLOT PRINT QUEUE TPUNCH	Output the specified text (a) in the output header page.
/NOTITLE	DIRECT	Do not output page headers. Complement of /TITLES. This is the default condition when outputting to the terminal.
/NOUNITS	DIRECT	Do not list the name of the actual disk unit; instead, list just the structure name. Complement of /UNITS. This is the default action.
/NOWRITE	SETSRC	Do not allow writing on the file structure for this job (i.e., the file structure is read only).



Switch	Applicable Commands/ Programs	Meaning
/NULL	CPUNCH PLOT PRINT QUEUE TPUNCH	Do not output an error message if there are no files in the request and do not create a queue entry. This is assumed at KJOB time.
/O	FAILSA          FILEX	<p>Enable the user to save and restore files to a user's area other than his own. This switch sets the destination project-programmer number to the specified number. The format is as follows:</p> <p style="text-align: center;">*/O xx,yyy&lt;CR&gt;</p> <p>The new value is retained until the next /O switch. If this switch is not given, the project-programmer number of the user running the job is used.</p> <p>Use the old DEC PDP-6 DECTape format.</p>
/OKBINARY	PRINT QUEUE	Print files whose extensions include binary information. Otherwise, files with extensions .SAV, .SHR, .LOW, .REL, and .HGH will not be in print queues.
/OKNONE	CPUNCH DIRECT PLOT PRINT QUEUE TPUNCH	Do not produce message if no files match the wildcard construction. However, a totally null queue request produces a fatal error message.
/OLD	COMPILE DEBUG EXECUTE LOAD	Run the appropriate language translator from the system library of old programs (device OLD:) which resides on the disk area [1,3]. If the translator does not exist on device OLD:, try to obtain it from device SYS:.
/OPTION:name	DIRECT QUEUE	Read the user's option file (DSK:switch.INI[ , ] /PHYSICAL) to determine the user's specified switch defaults for DIRECT. The name appearing as the value of the switch is the pointer to the line to read in the file.

Switch	Applicable Commands/ Programs	Meaning
/OUTPUT:n	QUEUE SUBMIT	<p>Cause job to terminate with a /Z:n to KJOB. n is from 0 to 4:</p> <p>n = 0 Suppress all normal queuing performed at LOGOUT time.</p> <p>n = 1 Queue only the log file.</p> <p>n = 2 Queue only the log file and spooled output (e.g., *.LPT).</p> <p>n = 3 Queue the log file, spooled output, and *.LST files.</p> <p>n = 4 Queue the log file, spooled output, *.LST files, and any requests deferred to LOGOUT time (default).</p>
/P	FAILSA	<p>Print a directory of all files on the tape on logical device LST. The directory is listed by project-programmer number and includes the number of blocks allocated per file, a running total for each area, and creation time/access date information. If the logical device LST is assigned to the disk or a DECtape, the filename FAILSA.DIR is given to the directory file.</p>
	FILEX	<p>Preserve the scratch file on the disk (resulting in quick processing). This file is deleted after processing is completed.</p>
	GLOB	<p>List all routines that define a symbol to have the same value. The routine that defines the symbol first is listed followed by a plus (+) sign. Subsequent routines that define the symbol are listed preceded by a plus sign.</p>
/PAGES:n	QUEUE PRINT SUBMIT	<p>Use n (decimal) as the maximum number of pages of output that the job can print. If the entire switch is omitted, the maximum is 200 pages; if only the value is omitted, the maximum is 2000 pages.</p>
/PAPER:x	QUEUE	<p>Identical to /PUNCH:x, /PRINT:x, /TAPE:x, or /PLOT:x.</p>

Switch	Applicable Commands/ Programs	Meaning
/PARITY: EVEN /PARITY ODD	DIRECT	Specify the parity to be used when reading a magnetic tape. The default is ODD.
/PAUSE	DISMOUNT MOUNT	Notify the user before sending the message to the operator for a request. The user can then abort the command if desired.
/PHYSICAL	CPUNCH DIRECT PLOT PRINT QUEUE SUBMIT TPUNCH	Suppress logical device names for the specified device.
/PLOT:ASCII	PLOT QUEUE	Plot the file on ASCII mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.
/PLOT:BINAR	PLOT QUEUE	Plot the file in binary mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.
/PLOT:IMAGE	PLOT QUEUE	Plot the file in image mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.
/PRINT:ARROW	PRINT QUEUE	Convert all control characters to up-arrow format except 011-015 and 020-024. This is the default condition.
/PRINT:ASCII	PRINT QUEUE	Send the file to the line printer with no changes.
/PRINT:OCTAL	PRINT QUEUE	Print the file in octal.

Switch	Applicable Commands/ Programs	Meaning
/PRINT: SUPPRESS	PRINT QUEUE	Suppress all character-control characters except for ASCII code characters LF and CR; this switch implies the use of the /PRINT:ARROW. Equivalent to operator command to spooler (SUPPRESS).
/PRIORITY:n	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Give the specified external priority (n = 0 to 62) to the request. A larger number is greater priority. The default is 10 if no switch is given, and 20 if switch is given without the value.
/PROTECT:nnn	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Specify a protection nnn (in octal) for this job or queue entry. If the switch, or the value of the switch is omitted, the standard protection is assumed.
/PROTECTION: nnn	DIRECT	Give the output file the protection nnn (in octal).
/PUNCH:026	CPUNCH QUEUE	Punch files in 025 Hollerith code. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:ASCII	CPUNCH QUEUE	Punch files in ASCII card code. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:BINARY	CPUNCH QUEUE	Punch files in binary card format. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:D029	CPUNCH QUEUE	Punch files in the old DEC 029 card code. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:IMAGE	CPUNCH QUEUE	Punch files in image mode. If the /PUNCH switch is not given, the file will be punched according to the data mode of the file.

Switch	Applicable Commands/ Programs	Meaning
/Q	FAILSA	Disable the detailed progress reports given by FAILSA on a single-user restore. Complement of /T. /Q is assumed unless changed by a /T switch.
	FILCOM	Print the message ?FILES ARE DIFFERENT when the files are different, but do not list the differences.
	FILEX	DEctape processing specifier indicating quick processing. This switch causes an input or output DEctape to be processed quickly by creating a scratch file on the disk. This file is deleted after processing is completed.
	GLOB	Suppress the listing of subsequent definers that result from the /P switch.
/R	FAILSA	Restore all files from the tape to the disk. The user must be logged in under [1,2] to use this feature.
	FILEX	DEctape processing specifier indicating reuse. This switch reuses a scratch file preserved by a /P switch in a previous command.
	GLOB	List only relocatable symbols.
/REL	COMPILE DEBUG EXECUTE LOAD	Use the existing .REL files although a newer source file may be present.
/REMOVE	CPUNCH PLOT PRINT QUEUE TPUNCH	Remove the file from the queue. This switch is valid only with the /MODIFY switch and can be used to remove a previously submitted file as long as that file's processing has not been started.
	DISMOUNT	Notify the operator to physically remove disk packs, tapes or cards.
/REPORT:code	PRINT QUEUE	Print the specified report within a COBOL report file. Code can be up to 12 characters in length.

Switch	Applicable Commands/ Programs	Meaning
/RESTART: 0 or 1	QUEUE SUBMIT	A value of 0 (default) means the job cannot be requeued or restarted by the operator after a system crash. A message is sent to the job' log file. A value of 1 means the job will be requeued or restarted. The job should not be restartable if there are changes to the permanent file directory.
/REWINDS	DIRECT.	Rewind the magnetic tape before and after reading it. Complement of /NOREWINDS. This is the default action.
/RONLY	MOUNT	Read only, means the same as /WLOCK.
/RUN:file spec	DIRECT QUEUE	Run the specified program when this command is finished.
/RUNOFFSET:n	DIRECT QUEUE	Run the program specified with /RUN with an offset of n. If the switch is omitted, the default is 0; if the switch is given without a value for n, the default value for n is 1.
/S	FAILSA	Save disk files on the magnetic tape. Up to 16 arguments can follow this switch, separated by spaces and/or tabs. These arguments are names of file structures or disk devices from which files are to be taken. If no arguments are specified, the entire disk is saved; in this case the user must be logged in under [1.2].
	FILCOM	Ignore spaces and tabs.
	FILEX	Use simple block (SBLK) file format. The default output extension is .BIN.
	GLOB	List symbols with non-conflicting values that are defined in more than one program.
/SCAN	SETSRC	Set the scan switch for the directory path.
/SEARCH	COMPILE DEBUG EXECUTE LOAD	Load the files in library search mode. This switch is identical to that of the /LIBRARY switch.

Switch	Applicable Commands/ Programs	Meaning
/SELF	COMPILE DEBUG EXECUTE LOAD	Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for an individual who keeps a private copy of a translator in his own disk area.
/SEQUENCE:n	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Specify a sequence number to aid in identifying a request to be modified or deleted.
/SINCE:t	CPUNCH DIRECT PLOT PRINT QUEUE TPUNCH	Queue only the files with creation dates after the specified time t where t is in the form dd-mm-yy hh mm.
/SINGLE	MOUNT	Permit only this job to access files on the structure (single access); file protection is enforced for it. disk only.
/SLOW	DIRECT	Output a full listing that includes the filename, extension, length in blocks written, protection, creation date and time, access date, structure name, and directory name.
/SNOBOL	COMPILE DEBUG EXECUTE LOAD	Compile the file with SNOBOL. Assumed for files with a .SNO extension.
/SORT	DIRECT	List both the file structure name (if there is more than one or if the files are on magnetic tape) and directory name (if a wildcard directory is given) on each line instead of only on the first line in which they change. Complement of /NOSORT.
/SPACING: DOUBLE	PRINT QUEUE	Double-space the output lines.

Switch	Applicable Commands/ Programs	Meaning
/SPACING: SINGLE	PRINT QUEUE	Single-space the printed lines. This is the default condition.
/SPACING: TRIPLE	PRINT QUEUE	Triple-space the printed lines.
/START:n	CPUNCH PLOT PRINT QUEUE SUBMIT TPUNCH	Start on line n of the file. If the switch, or the value of the switch is omitted, the default line at which to start is the first line.
/STRS	CPUNCH PLOT PRINT QUEUE TPUNCH	Search for the file on all structures in the search list and take each occurrence. The default is to take just first occurrence of this file.
/SUMMARY	DIRECT	Output only the summary line that indicates the total number of blocks and files. Note a /FAST/SUMMARY lists a /FAST listing followed by the summary.
/SYS	COMPILE DEBUG EXECUTE LOAD  SETSRC	Run the appropriate processor from the system library (device SYS:) area of [1,4]. This is the default condition.  Add the SYS specification to the user's DSK specification.
/T	DTCOPY  FAILSA	Write the bootstrap loader in blocks 0, 1 and 2 of the output DEctape. This switch accepts, as input from the terminal, a core bank, or offset. The loader is offset and then written on the tape.  Tell the user, on a single-user restore, the user area that has been found on the tape, the file that has been selected for transfer to the disk, when the transfer has begun, and when the transfer is completed. The /T switch takes effect on the file for which it appears and ends when a /Q switch is given.



Switch	Applicaote Commands/ Programs	Meaning
	FILEX	Use the normal PDP-10 directory DECtape format.
	FUDGE2	Skip to the logical end of the tape on a magnetic tape.
/TAG:xxx	QUEUE SUBMIT	Start at the statement labelled xxx (up to 5 characters) of the control file. Equivalent to GOTO xxx at the beginning of the control file.
/TAPE:ASCII	QUEUE TPUNCH	Punch the paper tape in ASCII code. If the /TAPE switch is not given, the files are punched according to the data mode of the file.
/TAPE:BINARY	QUEUE TPUNCH	Punch the paper tape in binary mode. If the /TAPE switch is omitted, the files are punched according to the data mode of the file.
/TAPE:IBINARY	QUEUE TPUNCH	Punch the paper tape in image - binary mode. If the /TAPE switch is omitted, the files are punched according to the data mode of the file.
/TAPE:IMAGE	QUEUE TPUNCH	Punch the paper tape in image mode. If the /TAPE switch is omitted, the files are punched according to the data mode of the file.
/TIME:hhmmss	QUEUE SUBMIT	Specify the central processor time limit for the job. If no switch is specified, the limit is 5 minutes; if the switch is specified without a value, the limit is 1 hour.
/TITLES	DIRECT	Output a heading on each page consisting of a label for each column, date, time, and page number. Complement of /NOTITLES.
/TPLOT:n	QUEUE SUBMIT	Use n (decimal minutes) as the maximum amount of plotting time allowed for the job. If the switch is omitted, no plotter time is allowed. If the value is omitted but the switch is given, the maximum plotter time is a function of file length.

Switch	Applicable Commands/ Programs	Meaning
/U	FAILSA	Saves, on the magnetic tape, the disk files in the UFD of the user last named in a /G switch, or if no /G switch has been given, of the user currently logged in.
	FILCOM	Compare two files in update mode.
/nU	FILCOM	Specify the upper limit for a partial binary compare (n is an octal number). This switch, when used with the /nL switch, allows a binary file to be compared only within the specified limits.
/UNIQUE: 0 or 1	QUEUE SUBMIT	Run any number of Batch jobs under this project-programmer number at the same time, if 0. Runs only one Batch job at any one time, if 1 (default).
/UNIT	DIRECT	List the name of the actual disk unit on which the files are stored instead of the file structure name. Complement of /NOUNITS.
/UNPRESERVED	CPUNCH PLOT PRINT QUEUE TPUNCH	Output file only if not preserved.
/V	DTCOPY  FAILSA  FILEX	Verify the similarities of the two DECtapes by performing a word-by-word comparison and typing on the terminal the number of discrepancies discovered.  Generate a request to lock the job in core, or reset the request.  DECtape format specified as in PDP-11.
/VID:name	MOUNT	Pass a visual identification to the operator as a comment to assist him in identifying a particular unit to mount. The argument can be in one of two forms:



Switch	Applicable Commands/ Programs	Meaning
/X	FAILSA	Extract the project-programmer numbers from the tape and reproduce output similar to the original TTY output generated by FAILSA during a save. The output contains the structure names and the project-programmer numbers on the tape. It is placed on logical device LST, and if LST is assigned to the disk or DECTape, the filename FAILSA.DIR is given to the output.
	FILCOM	Expand SAV files before comparing them in binary mode. This action removes differences resulting from zero compression.
	GLOB	Do not print listing header when output device is not the terminal, and include listing header when it is the terminal. Without this switch, the header is printed on all devices except the terminal.
/Y	FAILSA	This switch is used to debug new features.
	FAILSA	This switch is used to debug new features.
	FILEX	DECTape processing specified as zero. This causes the appropriate format of a zeroed directory to be written on a DECTape output file.
	FUDGE2	Clear the directory of the output DECTape.
/ZDEFER	QUEUE	Create a new entry in a queue and defer it until LOGOUT; however, the deferred file is zeroed first so that all previous /DEFER requests from the current job are deleted.
/2	FAILSA	Set the magnetic tape density to 200 bpi.
/5	FAILSA	Set the magnetic tape density to 556 bpi.
/6	COPY	Look for the directory in PDP-6 format (i.e., in block one instead of block 144).
/8	FAILSA	Set the magnetic tape density to 800 bpi.

## INDEX

- \$-language card, 3-9
- + construction, 1-17
- /026 Switch, 3-12, 3-14, 3-16, 3-19, 3-23, 3-27, 3-34, 3-35, 3-39
- /2 Switch, 2-103
- /5 Switch, 2-103, 2-120
- /6 Switch, 2-77
- /8 Switch, 2-103
- <> construction, 1-18
- = construction, 1-17
- @ construction, 1-16
  
- A Command (SETSRC), 2-246
- /A Switch, 2-99, 2-107, 2-120, 2-135
- ABS Extension, A-1
- Absolute,
  - date argument, 1-12
  - time argument, 1-12
- Access a file structure, 2-166
- Access codes,
  - Square brackets in, 1-10
- Access date word,
  - Set, 2-100
- Access protection codes,
  - 1-10, 2-191
- /ACCESS Switch, 2-65
- Accessing the system, 1-7, 2-162
- Add a device, 2-244
- ADDRESS Command, 2-81
- Administrative file, 2-166
- Advance a magnetic tape, 2-99, 2-129
- /AFTER Switch, 2-32, 2-178, 2-185, 2-196, 2-268, 2-285, 3-29
  
- ALCFIL Command, 2-3
- ALCFIL Program, 2-9
- .ALG Extension, 2-19, A-1
- \$ALGOL Card, 3-10
- ALGOL Compiler, 2-19
- ALGOL Program,
  - Compile an, 3-10
- /ALGOL Switch, 2-20, 2-56, 2-92, 2-155
- ALL Command, 2-81
- /ALLOC Switch, 2-65
- Allocate,
  - Devices, 2-2, 2-11
  - Space, 2-3
  - Space for new file, 2-9
  - System facilities, 2-2
- Allocated,
  - Blocks, 2-210
- Allocated core,
  - Return, 2-146
- Allocation commands,
  - Facility, 2-1, 2-2
- Alter,
  - a search list, 2-246
- Access protection codes, 2-191
- Alternate entry point, 2-218
- Analysis,
  - Tape error, 2-166
- Analyze a file, 2-79
- APPEND Command, 2-81
- Argument,
  - Absolute date, 1-12
  - Absolute time, 1-12
  - Command, 1-7
  - Core, 2-224, 2-226
  - Relative date, 1-12
  - Relative time, 1-12
- ASCII,
  - Card codes, B-1
  - Code, 3-1
  - Processing, 2-120
  - Switch, 3-11, 3-13, 3-15, 3-17, 3-20, 3-33, 3-35, 3-38
- Assembler,
  - MACRO, 2-19
  - MACX11, 2-19
- Assign a device, 2-166

ASSIGN Command, 1-7, 2-2, 2-11  
 Asterisk (\*), 1-2, 1-3  
 Asterisk as wild character, 1-11  
 ATTACH Command, 2-7, 2-13  
 Attach,  
   a job, 2-216  
   Terminal to job, 2-13  
 AUTO Command (OPSER), 2-170  
 AUTOFORMAT Command, 2-81  
 Available devices,  
   Print, 2-221  
  
 /B Switch, 2-99, 2-107,  
   2-120, 2-129  
 .B10 Extension, 2-19, A-1  
 Backspace a magnetic tape, 2-99, 2-129  
 BACKSPACE Command, 2-3, 2-15  
 .BACTO Command, 3-41  
 Backwards,  
   Move a tape, 2-3  
   Space a tape, 2-3  
 BAERR Message Indicator, 3-48  
 BAOPR Message Indicator, 3-48  
 BASUM Message Indicator, 3-48  
 BATCH  
   Command interpreter, 1- 5  
   Components, 3-39  
   Control file commands, 3-39  
   Controller, 1-2, 1-5, 3-1, 3-2, 3-3, 3-8  
   Controller messages, 3- 48  
   Illegal monitor commands in, 3-4  
   Input queue, 1-6, 2-193, 2-194, 2-267  
   Interpreting monitor commands in, 3-2  
   Jobs, 3-1  
   Message identifier, 3-48  
   Output, 3-6  
   Sample jobs, 3-49  
   Submitting jobs to, 3-4  
 BATCON, 1-5, 3-1, 3-2  
 BCM Extension, A-1  
 BCP Extension, A-1  
 BDATE Message Identifier, 3-48  
 /BEFORE Switch, 2-32, 2- 65,  
   2-178, 2-185, 2-196, 2- 285  
 /BEGIN Switch, 2-32, 2- 178,  
   2-185, 2-196, 2-285  
 /BIN Switch, 2-20, 2-56,  
   2-92, 2-155  
 Binaries,  
   Produce relocatable, 2- 19

Binary,  
   Code, 3-1  
   Compare, 2-107  
   Compare output, 2-109  
   Processing, 2-10  
   Processor switch, 1-20  
   Read files in, 2-134  
   Relocatable modules, 2- 126  
 /BINARY Switch, 3-17, 3- 21  
 Bits,  
   Privilege, 2-48  
 .BLI Extension, 2-19, A-1  
 BLISS Program,  
   Compile a, 3-12  
 /BLISS Switch, 2-20, 2-56, 2-93  
 BLIS10 Compiler, 2-19  
 \$BLISS Card, 3-12  
 Block format,  
   Simple, 2-120  
 Blocks,  
   Allocated, 2-210  
   Number of, 2-210  
   Numbering, 2-129  
 Blocks read,  
   Disk, 2-48  
 /BLOCKS Switch, 2-65  
 Blocks written,  
   Disk, 2-48  
 Blocksize,  
   Magnetic tape, 2-230  
 BLOCKSIZE Command,  
   SET, 2-3, 2-230  
 Blocksize for mag tape unit, 2-3  
 Brackets in access codes,  
   Square, 1-10  
 BREAK Command,  
   SET, 2-231  
 Busy test, 2-228  
 BVERS Message Identifier, 3-48  
  
 C Command (SETSRC), 2-247  
 /C Switch, 2-76, 2-99,  
   2-107, 2-120  
 Card,  
   \$-language, 3-9  
   \$ALGOL, 3-10  
   \$BLISS, 3-12  
   \$COBOL, 3-14  
   \$DATA, 3-9, 3-16  
   \$DECK, 3-9

- \$DECK, 3-20
- \$DUMP, 3-23
- End of job, 3-6
- \$EOD, 3-23
- \$EOJ, 3-6, 3-10, 3-24
- \$ERROR, 3-24
- \$EXECUTE, 3-6, 3-7, 3-25
- \$F40, 3-25
- \$FORTRAN, 3-5, 3-7, 3-25
- Image of the \$EXECUTE, 3-7
- Image of the \$FORTRAN, 3-7
- Image of the \$JOB, 3-6
- \$INCLUDE, 3-27
- \$JOB, 3-5, 3-6, 3-28
- \$MACRO, 3-32
- \$MODE, 3-19, 3-34
- \$NOERROR, 3-24
- \$PASSWORD, 3-5, 3-28, 3-36
- \$RELOCATABLE, 3-9, 3-36
- \$SEQUENCE, 3-37
- \$SNOBOL, 3-37
- CARD CODES,
  - ASCII, B-1
  - DEC-026, B-1
  - DEC-029, B-1
- Card deck,
  - Sample, 1-3
- Card input,
  - Terminate, 3-23
- Card punch,
  - Output queue, 2-31
  - Queue, 2-193
- Card-reader spooler, 2-234
- Cards,
  - Control, 3-5
  - Dollar sign in control, 3-9
  - Execute a program from, 3-16
  - File, 3-9
  - SPRINT-10 Control, 3-9
- /CARDS Switch, 2-196, 2-268, 3-29
- Cards to a file,
  - Copy, 3-20
- CATEGORY Command, 2-82
- CCONT Command, 2-6, 2-7, 2-40
- CCONTINUE Command, 2-40
- CDP Command, 2-194
- CDPSPL Command, 3-1
- CDR,
  - Set filename for, 2-234
- CDR Command,
  - SET 2-3, 2-234
- Central processor time limit, 2-250
- Central station message, 2-228
- \$CERR, 3-40
- Change,
  - CPUS, 2-235
  - File specifications, 2-4, 2-219
  - Interpretation mode, 3-34
  - Logical station, 2-161
- Character,
  - Asterisk as wild, 1-11
  - Control-O Special, 1-4
  - Control-R Special, 1-5
  - Control-T Special, 1-5
  - Control-U Special, 1-4
  - Percent (%), 1-24
  - Question mark as wild, 1-11
  - Ready, 1-3
- Character-oriented editor, 2-3
- Characteristics,
  - Terminal, 2-252
- Characters,
  - Special, 1-2, 1-4
- Charge number, 2-48
- /CHARGE Switch, 2-196
- Check for global symbols, 2-134
- /CHECK Switch, 2-72, 2-166, 2-167
- /CHECKSUM Switch, 2-65
- .CHKPNT Command, 3-41
- Clear,
  - A DEctape directory, 2-11, 2-76, 2-129, 2-214
  - A device directory, 2-297
  - A directory, 2-297
  - Logical device names, 2-54
  - User core, 2-132, 2-212, 2-223
- CLOSE Command, 2-2, 2-17, 2-82
- :CLOSE Command (OPSER), 2-171
- Cluster count field, 2-9
- Cluster size, 2-9
- \$COBOL Card, 3-14
- COBOL Compiler, 2-19
- COBOL Program,
  - Compile a, 3-14
- /COBOL Switch, 2-20, 2-56, 2-93, 2-156
- Codes,
  - ASCII, 3-1
  - ASCII Card, B-1
  - Binary, 3-1
  - DEC-026 Cards, B-1
  - DEC-029 Cards, B-1
  - Hollerith, 3-1
- Codes,
  - Access protection, 1-10
  - Redefining protection, 1-10
  - Square brackets in access, 1-10

Colon (:) in filename, 1-10

Command,

ADDRESS, 2-81  
ALCFIL, 2-3  
ALL, 2-81  
APPEND, 2-81  
ASSIGN, 1-7, 2-2, 2-11  
:AUTO (OPSER), 2-170  
AUTOFORMAT, 2-81  
BACKSPACE, 2-3, 2-15  
.BACKTO, 3-41  
CATEGORY, 2-82  
CCONT, 2-6, 2-7  
CCONTINUE, 2-40  
.CHKPNT, 3-41  
:CLOSE (OPSER), 2-171  
CLOSE, 2-2, 2-17, 2-82  
COMPILE, 2-5, 2-19, 3-5  
CONT, 2-6  
:CONTINUE (OPSER), 2-171  
CONTINUE, 2-26  
COPY, 2-4, 2-27  
CORE, 2-2, 2-29  
CPUNCH, 2-4, 2-31  
CREATE, 2-3, 2-37  
CREF, 2-5, 2-38  
CSTART, 2-6, 2-7, 2-40  
:CURRENT (OPSER), 2-171  
D, 2-6, 2-43  
:DAYTIME (OPSER), 2-171  
DAYTIME, 2-8, 2-45  
DCORE, 2-6, 2-46  
DDT, 2-6, 2-52  
DEASSIGN, 1-7, 2-2, 2-54  
DEBUG, 2-5, 2-55  
:DEFINE (OPSER), 2-171  
DELETE, 2-4, 2-61  
DEPOSIT, 2-6, 2-43  
DETACH, 2-7, 2-63  
:DEVICE (OPSER), 2-171  
DIRECT, 2-4, 2-64, 2-210  
DISMOUNT, 2-2, 2-72  
DSK, 2-8, 2-74  
DTCOPY, 2-76  
DUMP, 2-6, 2-79  
E, 2-6, 2-89  
EDIT, 2-3, 2-90  
EJECT, 2-82  
EOF, 2-4, 2-91  
.ERROR, 3-42  
:ERROR (OPSER), 2-171  
:EXIT (OPSER), 2-171  
EXAMINE, 2-6, 2-89  
EXECUTE, 2-5, 2-92

EXIT, 2-82  
FAILSA, 2-4  
FILCOM, 2-4  
FILE, 2-4, 2-115  
FILEX, 2-4  
FINISH, 2-2, 2-122  
:FREE (OPSER), 2-171  
FUDGE, 2-5, 2-124  
FUDGE2, 2-5  
GET, 2-6, 2-132  
GLOB, 2-4  
.GOTO, 3-42  
GRIPE, 2-7  
HALT, 2-6, 2-139  
HELP, 2-82, 2-140  
:HELP (OPSER), 2-171  
.IF, 3-43  
INITIA, 2-1, 2-144  
INPUT, 2-82  
IRADIX, 2-82  
JCONT, 2-6  
:JCONT (OPSER), 2-171  
JCONTINUE, 2-145  
JUSTIFY, 2-83  
KJOB, 2-7, 2-146  
:KJOB (OPSER), 2-171  
:KILL (OPSER), 2-171  
:KSYS (OPSER), 2-171  
LABEL, 2-4, 2-151  
LEFTMARGIN, 2-83  
LINEPAGE, 2-83  
LIST, 2-4, 2-153  
LOAD, 2-5, 2-155  
LOCATE, 2-2, 2-161  
LOGIN, 2-1, 2-162  
:LOGIN (OPSER), 2-171  
MAKE, 2-3, 2-164  
MODES, 2-83  
MOUNT, 2-2, 2-166  
:MSGLVL (OPSER), 2-171  
.NOERROR, 3-43  
.NOOPERATOR (BATCH), 3-4, 3-44  
.OPERATOR (BATCH), 3-4, 3-44  
OPSER, 2-7  
OPTION, 2-84  
ORADIX, 2-84  
OUTPUT, 2-84  
PJOB, 2-8, 2-174  
PLEASE, 2-7, 2-175  
PLOT, 2-4, 2-177  
PRESERVE, 2-4, 2-183  
PRINT, 2-4, 2-184  
Program examination, 2-6  
PROTECT, 2-4, 2-191



PUNCH, 2-4, 2-284  
 QUEUE, 2-4, 2-193  
 :QUEUE (OPSER), 2-172  
 QUOLST, 2-8  
 R, 2-6, 2-212  
 REASSIGN, 2-2, 2-214  
 REATTA, 2-7  
 REENTER, 2-6, 2-218  
 RENAME, 2-4, 2-219  
 .REQUEUE, 3-45  
 RESOURCES, 2-8, 2-221  
 :RESOURCES (OPSER), 2-172  
 :RESTRICT (OPSER), 2-172  
 :REVIVE (OPSER), 2-172  
 .REVIVE, 3-46  
 REWIND, 2-4, 2-222  
 RIGHTMARGIN, 2-84  
 RUN, 2-6, 2-85, 2-223  
 SAVE, 2-5, 2-225, D-1  
 SCHED, 2-8, 2-227  
 SEND, 2-7, 2-228  
 :SEND (OPSER), 2-172  
 :SET (OPSER), 2-172  
 SET BLOCKSIZE, 2-3, 2-230  
 SET BREAK, 2-231  
 SET CDR, 2-3, 2-234  
 SET CPU, 2-3, 2-235  
 SET DENSITY, 2-3, 2-237  
 SET DSKFUL, 2-6, 2-238  
 SET DSKPRI, 2-3, 2-239  
 SET HPQ, 2-3, 2-240  
 SET PHYSICAL GUIDELINE, 2-242  
 SET PHYSICAL LIMIT, 2-242  
 SET SPOOL, 2-3, 2-244  
 SET TIME, 2-8  
 SET TTY, 2-3  
 SET VIRTUAL LIMIT, 2-258  
 SET WATCH, 2-8  
 SETSRC, 2-8  
 .SILENCE, 3-46  
 :SILENCE (OPSER), 2-172  
 SKIP, 2-4, 2-262  
 :SLOGIN (OPSER), 2-172  
 SSAVE, 2-5, 2-264, D-1  
 START, 2-6, 2-266  
 :STOP (OPSER), 2-172  
 SUBMIT, 2-4, 2-267  
 SUPERSEDE, 2-85  
 SYFILE, 2-85  
 SYSTAT, 2-8, 2-273  
 :SYSTAT (OPSER), 2-172  
 TDUMP, 2-85  
 TECO, 2-3, 2-280  
 TIME, 2-8, 2-146, 2-282

TITLE, 2-85  
 :TLOG (OPSER), 2-173  
 TPUNCH, 2-4, 2-284  
 :TSILENCE (OPSER), 2-173  
 :TTYTST (OPSER), 2-173  
 TYPE, 2-5, 2-85, 2-290  
 UNLOAD, 2-5, 2-291  
 :UNRESTRICT (OPSER), 2-173  
 USESTAT, 1-5, 2-8, 2-292  
 VERSION, 2-8, 2-293  
 :WHAT (OPSER), 2-173  
 WHERE, 2-8, 2-296  
 WIDTH, 2-85  
 XTRACT, 2-86  
 ZERO, 2-5, 2-297  
 Commands,  
   Arguments to, 1-17  
   Batch Control File, 3-39  
   COMPILE-Class, 1-15  
   Continuing, 1-6  
   Control, 1-2  
   Facility allocation, 2-1, 2-2  
   File manipulation, 2-3  
   Formats of, 1-6  
   Indirect, 1-16  
   Interpreting Batch Monitor, 1-5, 8-2  
   Job information, 2-8  
   Job initialization, 2-1  
   Job termination, 2-7  
   Language interpreter of, 1-2, 1-3  
   Multiple job control, 2-7  
   Processing, 1-3  
   Program control, 2-5  
   Program preparation, 2-5  
   Source file preparation, 2-3  
   Spaces in, 1-6  
   System information, 2-8  
   TABS in, 1-6  
   Terminating, 1-6  
   Unique names of, 1-6  
 Commands to DUMP, 2-81, 2-82,  
   2-84, 2-85, 2-86  
 Commands to stacker,  
   Control, 1-6  
 Comments, 1-6, 3-9  
 Communicating with system jobs, 1-1  
 Communication,  
   Interconsole, 2-7, 2-175, 2-228  
 Compare,  
   Binary, 2-107, 2-109  
   File versions, 2-4, 2-106  
   Source, 2-107, 2-109  
   Two DECTapes, 2-76  
 COMPIL Program, 2-124

- COMPIL-class Commands, 1-15
- Compilation listings, 2-19
- Compile,
  - A BLISS Program, 3-12
  - A COBOL Program, 3-14
  - A MACRO Program, 3-32
  - A SNOBOL Program, 3-37
  - An ALGOL Program, 3-10
- COMPILE Command, 2-5, 2-19, 3-5
- /COMPILE Switch, 2-20, 2-56, 2-93, 2-156
- COMPILE Switches, 2-20, 1-18
- Compiler,
  - ALGOL, 2-19
  - BLIS10, 2-19
  - COBOL, 2-19
  - FORTRAN, 2-19
  - SNOBOL, 2-19
- Components,
  - BATCH, 3-1
- Configuration table, 2-46
- Configuring the system, 1-1
- CONFIRM Dialogue, 2-146
- Construction,
  - +, 1-17
  - <>, 1-18
  - =, 1-17
  - @, 1-16
  - Wildcard, 1-11
- CONT Command, 2-6
- Contents operators, 2-86
- Continue,
  - A line, 1-6
  - A program, 2-26, 2-40, 2-6, 2-7
  - A job, 2-145, 2-6
  - command lines, 1-6
- CONTINUE Command, 2-26
- :CONTINUE Command (OPSER), 2-171
- Control,
  - Multiple job, 2-170
  - Transfer, 3-42
- Control bits,
  - Spooling, 2-48
- Control Cards,
  - Description of, 3-5
  - Dollar sign in, 3-9
  - SPRINT-10, 3-9
- Control commands, 1-2
- Control commands,
  - Multiple job, 2-7
  - Program, 2-5
  - To stacker, 1-6
- Control file,
  - Create a, 3-28
  - Description, 1-6
  - Interrupt, 3-39
  - Interrupt sequential reading of, 3-41
- Control File,
  - Commands to Batch, 3-39
  - Specifications, 2-194
  - Submitting a job with a, 3-8
  - User, 3-2
- Control over devices,
  - Operator, 2-166
- Control-C, 1-2, 2-26
- Control-O Special Character, 1-4
- Control-R Special Character, 1-5
- Control-T Special Character, 1-5
- Control-U Special Character, 1-4
- Controller,
  - Batch, 1-2, 1-5, 3-1, 3-8, 3-2, 3-3
- Controller Messages,
  - Batch, 3-48
- Conversational mode, 3-3
- /COPIES Switch, 2-32, 2-178, 2-185, 2-197, 2-285
- Copy,
  - Cards to a file, 3-20
  - DECTapes, 2-76
- COPY Command, 2-4, 2-27
- Core,
  - Arguments, 2-224
  - Clear user, 2-132, 2-212, 2-223
  - Examine a location in, 2-89
  - Lock job in, 2-103
  - Modify amount of, 2-29
  - Return allocated, 2-146
  - Type amount of, 2-29
  - User's area of, 2-46
- CORE Command, 2-2, 2-29
- Core image file,
  - Load a, 2-132, 2-212, 2-223
  - Write a, 2-5, 2-46, 2-79
  - Write to a device, 2-225
- /CORE Switch, 2-197, 2-268, 3-29
- Count,
  - Dependency, 3-8
- Count field,
  - Cluster, 2-9
- Counter,
  - Program, 1-5
  - Response, 2-48
- CP Command (SETSRC), 2-247
- CPPL, 2-242
- CPU Command,
  - SET, 2-3, 2-235
- CPUNCH Command, 2-4, 2-31
- /CPUNCH, Switch, 3-21

CPUs,  
   Change, 2-235  
 Create,  
   A control file, 3-28  
   A library file, 2-124  
   A new file, 2-164  
 CREATE Command, 2-3, 2-37  
 /CREATE Switch, 2-33, 2-178, 2-185,  
   2-197, 2-247, 2-268, 2-286  
 Creation,  
   Open a file for, 2-37  
   Set date of, 2-100  
   UFD, 2-166  
 CREF Command, 2-5, 2-38  
 /CREF Switch, 2-21, 2-56,  
   2-93, 2-156, 3-27, 3-33  
 Cross-referenced,  
   Listings, 2-38  
   Symbols list, 2-134  
 CS Command (SETSRC), 2-247  
 CSTART Command, 2-6, 2-7, 2-40  
 .CTL, 2-194  
 .CTL Extension, 3-28  
 :CURRENT Command (OPSER), 2-171  
 Current job,  
   Detach, 2-7  
 Current,  
   Virtual page limit, 2-258  
   Physical page limit, 2-242  
 CVPL, 2-258

D Command, 2-6, 2-43  
 /D Switch, 2-100, 2-120  
 DAEMON Program, 2-46, 2-79  
 DAEMON-written File, 2-47  
 \$DATA Card, 3-9, 3-16  
 \$DATA Card Restrictions, 3-20  
 Data,  
   File, 1-6  
   Mode, 1-2  
   User file, 3-1  
 Date,  
   Set creation, 2-100  
   Type current, 2-8  
   Type the current, 2-45  
 Date argument,  
   Absolute, 1-12  
   Relative, 1-12  
 Date arguments, 1-12  
 Date word,  
   Set access, 2-100

Day,  
   Type the time of, 2-45  
 Day time,  
   Incremental, 1-5  
 DAYTIME Command, 2-8, 2-45  
 :DAYTIME Command (OPSER), 2-171  
 DCORE Command, 2-6, 2-46  
 DDBs, 2-46  
 DDT Command, 2-6, 2-52  
 /DDT Switch, 2-56  
 /DEADLINE Switch, 2-33, 2-179, 2-186, 2-197  
   2-268, 2-286, 3-29  
 DEASSIGN Command, 1-7, 2-2, 2-54  
 DEBUG Command, 2-5, 2-55  
 DEC-026 CARD CODES, B-1  
 DEC-029 CARD CODES, B-1  
 Deck,  
   Sample card, 1-3  
 Deck Card, 3-9  
 \$DECK Card, 3-20  
 DECsystem-10,  
   Functions of, 1-1  
 DECTape,  
   Rewind a, 2-222, 2-291  
   Unload a, 2-291  
 Default switches,  
   Override system, 1-14  
 /DEFER Switch, 2-198  
 :DEFINE Command (OPSER), 2-171  
 Delete,  
   a device, 2-244  
   a file from DECTape, 2-61  
   A file from disk, 2-61, 2-101  
   All files, 2-146  
   unprotected files, 2-146  
 DELETE Command, 2-4, 2-61  
 DENSITY Command,  
   SET, 2-3, 2-237  
 Density,  
   For mag tape unit, 2-3, 2-237  
   Set tape, 2-100, 2-103  
 /DENSITY Switch, 2-65  
 /DEPEND Switch, 2-198, 2-269, 3-30  
 Dependency,  
   Interjob, 3-8  
 Dependency count, 3-8  
 DEPOSIT Command, 2-6, 2-43  
 Deassociate a device from job, 2-122  
 Descriptor,  
   DUMP, 2-86  
 DETACH Command, 2-7, 2-63  
 Detach current job, 2-7, 2-13  
 /DETAIL Switch, 2-65

September 1974

- Determine,
  - Progress of job, 1-5
  - Station number, 2-296
- Device,
  - Add a, 2-244
  - Allocates a, 2-11, 2-2
  - Assign a, 2-166
  - Clear directory of a, 2-297
  - Data blocks, 2-46
  - Directory oriented, 1-10
  - Delete a, 2-244
  - Disassociate a job from, 2-122
- List of, 1-10
- Names, 1-7
- Stop a, 2-146
- Structure name, 2-48
- Write a core image to, 2-225
- :DEVICE Command (OPSER), 2-171
- Device names,
  - Clear logical, 2-54
  - Generic physical, 1-8
  - Logical, 1-7
  - Physical, 1-7
- Device pool,
  - Return devices to, 2-54
- Device pools,
  - Monitor, 2-72, 2-54, 2-146
- Device to monitor pool,
  - Return, 2-54, 2-146
- Devices,
  - Operator control over, 2-166
  - Pass, 2-214
  - Print available, 2-8, 2-221
  - Restricted, 2-2, 2-214
  - Spooling, 2-244
  - Unrestricted, 2-2, 2-214
- Dialogue,
  - CONFIRM, 2-146
- Dialogue mode,
  - Enter, 3-4, 3-44
  - Leave, 3-4, 3-44
- DIRECT Command, 2-4, 2-64, 2-210
- Directories,
  - Clear, 2-5
  - Clear DECTape, 2-11, 2-76, 2-129, 2-214
  - Clear device, 2-297
  - Entries, 2-64
  - List, 2-146
  - Names, 1-10
  - Of files on tape, 2-101
  - Of LST, 2-102
  - sub-file, 1-10
  - User file (UFD), 1-10, 2-166
- Directory-oriented devices, 1-10
- Disconnect a terminal, 2-7
- Disconnect job's terminal, 2-63
- Disk,
  - blocks read, 2-48
  - blocks written, 2-48
  - delete file from, 2-61, 2-101
  - incremental reads, 1-5
  - incremental writes, 1-5
  - performance statistics, 2-277
  - restore files on, 2-4
  - save files on, 2-4, 2-99, 2-102
  - space used, 2-210
  - transfer to/from tape, 2-115
  - usage for a job, 2-74
- DISMOUNT Command, 2-2, 2-72
- /DISPOSE Switch, 2-33, 2-179, 2-186, 2-198, 2-269, 2-286
- Dollar sign (\$)
  - in BATCH, 3-6
  - in control cards, 3-9
  - use of, 1-6
- DSK Command, 2-8, 2-74
- DSKFUL Command, SET, 2-6, 2-238
- DSKPRI Command, SET, 2-3, 2-239
- DTCOPY Command, 2-76
- DUMP,
  - Command, 2-6, 2-89
  - Commands to, 2-81, 2-82, 2-84, 2-85, 2-86
  - Descriptor, 2-86
  - Expressions to, 2-86
  - Program, 2-81
  - Symbols, 2-86
- Dump a file, 2-81
- \$DUMP Card, 3-23
- E Command, 2-6, 2-89
- /E Switch, 2-100, 2-120, 2-135
- EDIT Command, 2-3, 2-90
- Editing,
  - Open a file for, 2-90
  - Open file for, 2-280
- Editor, 2-280
- Editor,
  - Character-oriented, 2-3
  - Line-oriented, 2-3, 2-37, 2-90
  - Text, 2-164

- EJECT Command, 2-82
- Eliminate files, 2-100
- End of job, 3-10
- End of job card, 3-6
- End-of-file, 3-10
- End-of-file mark,
  - Write an, 2-91
- Enter dialogue mode, 3-4, 3-44
- Entries,
  - List directory, 2-64
- Entry point,
  - Alternate, 2-218
- \$EOD Card, 3-23
- EOF Command, 2-4, 2-91
- \$EOJ Card, 3-6, 3-10, 3-24
- EOT, 2-132
- /EOTS Switch, 2-66
- Equal sign (=), 1-2, 2-27
- %ERR, 3-40
- Erroneous symbols,
  - List, 2-135
- Error analysis,
  - Tape, 2-166
- \$ERROR Card, 3-24
- .ERROR Command, 3-42
- :ERROR Command (OPSER), 2-171
- Error messages, Chapter 4
- Error,
  - Recovery aid, 3-41
  - Recovery routines, 3-40
  - Reporting, 2-166
- Error Reporting,
  - SPRINT-10, 3-48
- /ERROR Switch, 3-30
- Errors,
  - Process, 3-43
- Examination Command,
  - Program, 2-6
- Examine a core location, 2-89
- EXAMINE Command, 2-6, 2-89
- Exclamation point (BATCH), 3-7
- \$EXECUTE Card, 3-6, 3-7, 3-25
- EXECUTE Command, 2-5, 2-92
- Execute,
  - a program from cards, 3-16
  - a program, 2-92
- Execution,
  - Continue job, 2-6
  - Continue program, 2-6, 2-7
  - program, 2-266
- EXIT Command, 2-82
- :EXIT Command (OPSER), 2-171
- Expressions to DUMP, 2-86
- Extensions, 2-19, A-1
- Extensions,
  - Filename, 1-10, 2-19, A-1
- /F Switch, 2-33, 2-100,
  - 2-119, 2-135, 2-179,
  - 2-186, 2-199, 2-269, 2-286
- /F10 Switch, 2-21, 2-57,
  - 2-93, 2-157
- .F4 Extension, 2-19
- SF40 Card, 3-25
- /F40 Switch, 2-22, 2-57
  - 2-93, 2-157
- Facilities,
  - Allocating system, 2-2
- Facility allocation
  - Commands, 2-1, 2-2
- FAILSA Command, 2-4
- FAILSAFE Program, 2-99
- /FAST Switch, 2-66
- Feature table, 2-46
- /FEET Switch, 2-199, 2-269, 3-31
- Field,
  - Cluster count, 2-9
- FILECOM Command, 2-4
- FILECOM Program, 2-106
- File (FUDGE2),
  - Master, 2-126
  - output, 2-126
  - transaction, 2-126
- File cards, 3-9
- FILE Command, 2-4, 2-115
- File Commands,
  - Batch Control, 3-39
- File directory,
  - User, 1-10
- File Directory, (UFD),
  - User, 2-166
- File for creation,
  - open a, 2-37
- File for editing,
  - open, 2-280
  - open a, 2-90
- File from dectape,
  - Delete a, 2-61
- File from disk,
  - delete, 2-101
  - delete a, 2-61
- File manipulation commands, 2-3
- File preparation commands,
  - source, 2-3

File specification,  
   Change, 2-4, 2-219  
   control, 2-194  
   listing, 2-194  
   log, 2-194  
 File structure,  
   access a, 2-166  
   remove a, 2-72  
 File structures,  
   print, 2-221  
 /FILE Switch, 2-186, 2-199, 2-286  
 File transfer program, 2-119  
 Filename,  
   Colon (:) in, 1-10  
   Octal representation of, 1-12  
   Period (.) in, 1-10  
 Filename extensions, 1-10  
 Filename for CDR,  
   set, 2-234  
 Files,  
   administrative, 2-166  
   allocate space for, 2-9  
   analyze a, 2-79  
   control, 1-6  
   copy cards to, 3-20  
   create,  
     control, 3-28  
     library, 2-124  
     new, 2-164  
   DAEMON-written, 2-47  
   data, 1-6  
   delete all 2-146  
   delete unprotected, 2-146  
   dump a, 2-81  
   eliminate, 2-100  
   format specifiers, 2-120  
   functions to, 2-115  
   indexed library, 2-129  
   interrupt control, 3-39  
   interrupt sequential reading of, 3-41  
   job log, 3-1, 3-2, 3-47  
   line-numbered, 2-90  
   load, 2-155  
   load .REL 2-92  
   log, 1-6  
   names, 1-10  
   option, 1-14  
   output source, 2-153  
   place output in log, 3-46  
   print a, 2-5, 2-90  
   produce source, 2-5  
   protect, 2-146  
   read binary, 2-134  
   rename a, 2-4, 2-183  
   restore a, 2-100 to 2-102  
   save, 2-4, 2-99 to 2-102, 2-146  
   specification, 1-10, 2-4, 2-194, 2-219  
   structure, 1-10, 2-48, 2-72,  
     2-166, 2-221  
   temporary, C-1  
   transfer, 2-4, 2-27, 2-115, 2-119  
   type a, 2-290  
   update a, 2-5  
   version, 2-4, 2-106  
   write a core image, 2-46  
 Files (BATCH),  
   submitting a job with, 3-6  
 Files on tape,  
   directory of, 2-101  
   /FILES Switch, 2-66  
 FILEX Command, 2-4  
 FILEX Program, 2-119  
 %FIN, 3-40  
 FINISH Command, 2-2, 2-122  
 Fixed symbols,  
   list, 2-135  
 .FOR Extension, 2-19  
 Format,  
   simple block, 2-120  
 Format specifiers,  
   DECtape, 2-119  
 Format specifiers,  
   file, 2-120  
 Formats,  
   command, 1-6  
   /FORMS Switch, 2-33, 2-187, 2-200  
   /FOROTS Switch, 2-56, 2-93, 2-156  
   /FORSE Switch, 2-56, 2-93, 2-156  
   \$FORTRAN Card, 3-5, 3-7, 3-25  
   \$FORTRAN Card,  
     image of the, 3-7  
   FORTRAN Compiler, 2-19  
   /FORTRAN Switch, 2-21, 2-56,  
     2-93, 2-156  
 Forward,  
   move a tape, 2-4  
 Free a terminal, 2-63  
 :FREE Command (OPSER), 2-171  
 Free system space, 2-210  
 FUDGE Command, 2-5, 2-124  
 /FUDGE Switch, 2-21, 2-57,  
   2-93, 2-124, 2-156  
 FUDGE2 Command, 2-5  
 FUDGE2 Program, 2-126  
 Functions,  
   of DECsystem-10, 1-1  
   of monitor, 1-1  
   to file, 2-115

- /G Switch, 2-76, 2-100, 2-120
- Gain access to system, 2-162
- Generic physical device names, 1-8
- GET Command, 2-6, 2-132
- GLOB Command, 2-4
- GLOB Program, 2-134
- Global symbols,
  - check for, 2-134
  - output, 2-135
- .GOTO Command, 3-42
- GRIPE Command, 2-7
- GRIPE Program, 2-138
- .GTCNF, 2-46
- .GTFET, 2-46
- GUIDELINE Command,
  - SET PHYSICAL, 2-242
  
- H Command (SETSRC), 2-247
- /H Switch, 2-76, 2-100,
  - 2-107, 2-108, 2-119, 2-135
- HALT,
  - start after a, 2-26
- HALT Command, 2-6, 2-139
- Header,
  - Listing, 2-136
- /HEADER Switch, 2-187, 2-200
- HELP Command, 2-82, 2-140
- :HELP Command (OPSER), 2-171
- /HELP Switch, 2-66, 2-72,
  - 2-167, 2-200
- High segment,
  - sharable, 2-264
- High-priority scheduler run queue, 2-240
- Hollerith Code, 3-1
- /HOLLERITH Switch, 3-31
- HPQ Command,
  - SET, 2-3, 2-240
- Hyphen (-), 1-6, 3-9
  
- /I Switch, 2-100, 2-120
- Identification,
  - Reel, 2-167
  - Terminal, 2-228
- Identifier,
  - BATCH Message, 3-48
  - BDATE Message, 3-48
  - BVERS Message, 3-48
  - DECtape, 2-4, 2-151
  - STDAT Message, 3-47
  - STERR Message, 3-47
  - STMSG Message, 3-47
  - STOPR Message, 3-47
  - STSUM Message, 3-47
  - IF Command, 3-43
- Illegal monitor commands in
  - BATCH, 3-4
- Image,
  - Load a core, 2-132, 2-212, 2-223
  - Write a core, 2-5, 2-79
- Image of the \$EXECUTE Card, 3-7
- Image of the \$FORTRAN Card, 3-7
- Image of the \$JOB Card, 3-6
- Image processing, 2-120
- /IMAGE Switch, 3-18
- Image to device,
  - Write a core, 2-225
- \$INCLUDE Card, 3-27
- Incremental,
  - day time, 1-5
  - disk reads, 1-5
  - disk writes, 1-5
  - run time, 1-5
- Indexed library files, 2-219
- Indicator,
  - BAERR Message, 3-48
  - BAOPR Message, 3-48
  - BASUM Message, 3-48
  - MONTR Message, 3-48
  - USER Message, 3-48
- Indirect commands, 1-16
- Information,
  - Job status, 2-8
  - Print status, 1-5
  - Status, 2-273, 2-292
- Information commands,
  - Job, 2-8
  - System, 2-8
- INITIA Commands, 2-1, 2-144
- Initialize
  - a job, 1-1
  - terminal, 2-144
  - Commands to, 2-1
- INP:, 2-194, 2-267
- Input,
  - Specifications, 2-194
  - Spooler, 1-5, 3-1

- Spooler Mode, 1-2
- Terminate, 2-122
- Terminate card, 3-23
- INPUT Command, 2-82
- Input line.
  - Retype, 1-5
  - Rescanning, 2-26
- Input queue.
  - Batch, 1-6, 2-193, 2-194, 2-267
- Input/output.
  - Terminate, 2-2, 2-17
- Interconsole communication.
  - 2-7, 2-175, 2-228
- Interjob dependency, 3-8
- Intermediate mode, 3-3
- Internal symbols table, 2-134
- Interpretation mode.
  - Change, 3-34
- Interpreter.
  - Batch command, 1-5, 3-2
  - Command language, 1-2
  - Monitor command language, 1-3
- Interpreting processor switches, 1-20
- Interrupt control file, 3-39
- Interrupt sequential reading of control file, 3-41
- IRADIX Command, 2-82

- Log file, 3-1, 3-2, 3-47
- Multiple control of, 2-170
- Number, 2-146
- Output, 3-46
- Priorities, 2-239
- Requeue a, 3-45
- Run time, 2-8
- Search list, 2-8, 2-246
- Scheduling, 3-2, 3-8
- Size, 1-5
- State, 1-5, 2-292, 2-259
- Status information, 2-8, 2-48
- Stop a, 2-6, 2-139
- Suppress all output from, 3-46
- Terminal number, 2-174
- Terminate a, 3-24
- Time limit for, 2-8
- Transfer a, 2-7
- JOB CAPACITY EXCEEDED Message, 1-2
- \$JOB Card, 3-5, 3-6, 3-28
- Job card.
  - End of, 3-6
- \$JOB Card.
  - Image of the, 3-6
- Jobs to BATCH.
  - Submitting, 3-4
- JUSTIFY Command, 2-83

- /J Switch, 2-100
- .JBOPC, 2-52, 2-218
- .JBPC, 2-26, 2-52, 2-139, 2-218
- JCONT Command, 2-6, 2-145
- :JCONT Command (OPSER), 2-171
- JCONTINUE Command, 2-145
- Jiffies.
  - Time limit in, 2-48
- Job.
  - Attach a, 2-216
  - Attach terminal to, 2-13
  - BATCH, 3-1, 3-4
  - Communicating with system, 1-1
  - Continue a, 2-145
  - Control commands a, 2-7
  - Disassociate a device from, 2-63, 2-122
  - Detach current, 2-7, 2-13
  - Determine progress of, 1-5
  - Disk usage for a, 2-74
  - End of a, 3-10
  - Information commands, 2-8
  - Initiating a, 1-1, 2-1
  - Lock in core, 2-103

- /K Switch, 2-101, 2-129
- /KA10 Switch, 2-22, 2-57, 2-93, 2-157
- Key.
  - RETURN, 1-6
  - RUBOUT, 1-4
- /KI10 Switch, 2-57, 2-93, 2-157
- :KILL Command (OPSER), 2-171
- /KILL Switch, 2-33, 2-179, 2-187, 2-200, 2-270, 2-286
- Kilo-core-ticks, 2-48
- KJOB Command, 2-7, 2-146
- :KJOB Command (OPSER), 2-171
- :KSYS Command (OPSER), 2-172
- /L Switch, 2-76, 2-101, 2-120, 2-135
- /nL Switch, 2-107, 2-108
- LABEL Command, 2-4, 2-151



Language interpreter,  
   Command, 1-2  
   Monitor command, 1-3  
 Leave dialogue mode, 3-4, 3-44  
 LEFTMARGIN Command, 2-83  
 Library file,  
   Create a, 2-124  
 Library files,  
   Indexed, 2-129  
 /LIBRARY Switch, 2-57, 2-93, 2-157  
 Limit,  
   Central processor time, 2-250  
   Current physical page, 2-242  
   Current virtual page, 2-258  
 LIMIT Command,  
   SET VIRTUAL, 2-258  
   SET PHYSICAL, 2-242  
 Limit for job,  
   Time, 2-8  
 Limit in jiffies,  
   Time, 2-48  
 /LIMIT Switch, 2-34, 2-180,  
   2-187, 2-201, 2-287  
 Line,  
   continuation, 1-6  
   Rescanning input, 2-26  
   Retype input, 1-5  
   Terminating a command, 1-6  
 Line printer,  
   Write to, 2-4  
 Line printer output queue,  
   2-184, 2-193, 3-4  
 Line-numbered file, 2-90  
 Line-oriented editor, 2-3  
   2-37, 2-90  
 LINED,  
   Run, 2-3, 2-37, 2-90  
 LINEPAGE Command, 2-83  
 /LINK Switch, 2-57, 2-93, 2-157  
 LINK-10,  
   Switch specification in, 1-25  
 LINK-10 program, 1-24  
 LINK-10 Switches,  
   Passing, 1-25  
 Linking-loaders,  
   Switches to the, 1-24  
 List,  
   Alter a search, 2-246  
   Cross-referenced symbols, 2-134  
   Devices, 1-8  
   Directories, 2-146  
   Directory entries, 2-64  
   Erroneous symbols, 2-135  
   fixed symbols, 2-135  
   Job search, 2-8, 2-246  
   Manipulate search, 2-246  
   Nonrelocatable symbols, 2-135  
   of Processor switches, 1-20  
   of Quotas, 2-166  
   of symbols, 2-134  
   Relocatable symbols, 2-135  
   source files, 2-153  
   System search, 2-8, 2-246  
 LIST Command, 2-4, 2-153  
 /LIST Switch, 2-22, 2-34,  
   2-58, 2-66, 2-93, 2-157, 2-180,  
   2-187, 2-201, 2-270, 2-287  
 Listing file specification, 2-194  
 Listing header, 2-136  
 Listings,  
   Compilation, 2-19  
   Cross-reference, 2-38  
 /LMAP Switch, 2-58, 2-95, 2-157  
 Load  
   a core image, 2-132, 2-212, 2-223  
   files, 2-155  
   .REL files, 2-92  
 LOAD Command, 2-5, 2-155  
 Loader program, 1-24  
 /LOADER Switch, 2-58, 2-95, 2-157  
 LOADER Switches,  
   Passing, 1-24  
 Loading the system, 1-1  
 LOCATE Command, 2-2, 2-161  
 /LOCATE Switch, 3-31  
 Location,  
   Examine a core, 2-89  
 Lock a job in core, 2-103  
 .LOG Extension, 3-28  
 Log file,  
   Job, 1-6, 3-1, 3-2, 3-47  
   Place output in, 3-46  
   specification, 2-194  
 /LOG Switch, 2-187, 2-201  
 Logged-input quota, 2-8  
 Logging on the system, 1-7  
 Logical device names, 1-7  
 Logical device names,  
   Clear, 2-54  
 Logical station, Change, 2-161  
 LOGIN Command, 2-1, 2-162  
 :LOGIN Command (OPSER), 2-172  
 ?LOGIN PLEASE message, 1-4  
 Logout, 2-146  
 Look for trailer record, 2-100  
 .LPT, 2-194

September 1974

LPT:, 2-194  
 LPTSPL, 3-1, 3-4  
 LST,  
     directory of, 2-102

M Command (SETSRC), 2-247  
 /M Switch, 2-101, 2-119, 2-135  
 .MAC Extension, 2-19  
 MACRO Assembler, 2-19  
 \$MACRO Card, 3-32  
 Macro program,  
     Compile a, 3-32  
 /MACRO Switch, 2-22, 2-58  
     2-95, 2-157  
 MACX11 Assembler, 2-19  
 /MACX11 Switch, 2-58, 2-95, 2-158  
 Magnetic tape,  
     Advance a, 2-99, 2-129  
     Backspace a, 2-99, 2-129  
     Density for, 2-237  
     Move a, 2-262  
     Rewind, 2-103  
     Rewind a, 2-222, 2-291  
     Unload a, 2-291  
 Magnetic tape blocksize, 2-230  
 Maintain system queues, 3-2,  
 MAKE Command, 2-3, 2-164  
 Manager,  
     Queue, 3-1, 3-2  
 Manipulate search list, 2-246  
 Manipulation commands,  
     File, 2-3  
 /MANTIS Switch, 2-22, 2-58  
     2-95, 2-158  
 /MAP Switch, 2-58, 2-95  
     2-158, 3-8  
 /MARKS Switch, 2-66  
 Masks,  
     Password, 2-7  
 Master File (FUDGE2), 2-126  
 Media error reporting, 2-166  
 Message,  
     Central station, 2-228  
     job capacity exceeded, 1-2  
 Message,  
     ?LOGIN PLEASE, 1-4  
     ?NO CORE ASSIGNED, 1-4  
     ?TIME LIMIT EXCEEDED, 3-44  
 Message Identifier,  
     BATCH, 3-48  
     BDATE, 3-48  
     BVERS, 3-48  
     STDAT, 3-47  
     STERR, 3-47  
     STMSG, 3-47  
     STOPR, 3-47  
     STSUM, 3-47  
 Message Indicator,  
     BAERR, 3-48  
     BAOPR, 3-48  
     BASUM, 3-48  
     MONTR, 3-48  
     USER, 3-48  
 Messages,  
     Batch Controller, 3-48  
     Error, Chapter 4  
     Sending, 2-7  
     SPRINT-10, 3-47  
 Mode,  
     Change interpretation, 3-34  
     Conversational, 3-3  
     Data, 1-2  
     Dialogue, 3-4  
     Enter dialogue, 3-4, 3-44  
     Input spooler, 1-2  
     Intermediate, 3-3  
     Leave dialogue, 3-4, 3-44  
     Monitor, 1-2  
     Terminal, 1-2  
     User, 1-2  
 \$MODE Card, 3-19, 3-34  
 MODES Command, 2-83  
 Modify amount of core, 2-29  
 /MODIFY Switch, 2-34, 2-180  
     2-187, 2-201, 2-270, 2-287  
 Modules,  
     Relocatable binary, 2-126  
 Monitor,  
     Batch Commands, 3-2  
     Command language interpreter, 1-3  
     Device pools, 2-72  
     Functions of, 1-1  
     Mode, 1-2  
 MONTR Message Indicator, 3-48  
 MOUNT Command, 2-2, 2-166  
 Move  
     a mag tape backwards, 2-15  
     a magnetic tape, 2-262  
     a tape backwards, 2-3  
 :MSGLVL Command (OPSER), 2-172  
 /MULTI Switch, 2-167  
 Multiple  
     job control, 2-170  
     job control commands, 2-7  
 Multiprogramming, 1-1

/N Switch, 2-76, 2-101, 2-135  
 /NAME Switch, 3-31  
 Names,  
   Clear logical device, 2-54  
   Command, 1-6  
   Device, 1-7  
   Device Structure, 2-48  
   Directory, 1-10  
   File structure, 1-10, 2-48  
   Generic physical device, 1-8  
   Logical device, 1-7  
   Obtain program, 1-5  
   Physical device, 1-7  
   Queue, 2-194  
   Unique command, 1-6  
   User program, 2-48  
 /NEW Switch, 2-22, 2-34,  
   2-58, 2-95, 2-158,  
   2-180, 2-187, 2-201  
   2-248, 2-270, 2-287  
 ?NO CORE ASSIGNED message, 1-4  
 /NOBIN Switch, 2-23, 2-58, 2-96, 2-158  
 /NOCHECKSUM Switch, 2-66  
 /NOCOMPILE Switch, 2-23, 2-59, 2-96, 2-158  
 /NOCREATE Switch, 2-247  
 /NODETAIL Switch, 2-66  
 /NOEOTS Switch, 2-66  
 \$NOERROR Card, 3-24  
 .NOERROR Command, 3-43  
 /NOLIB Switch, 2-248  
 /NOLIST Switch, 2-23, 2-59  
   2-96, 2-158, 3-11, 3-13  
   3-27, 3-33  
 /NOMANTIS Switch, 2-23, 2-59, 2-96, 2-158  
 /NOMARKS Switch, 2-66  
 /NONEW Switch, 2-248  
 Nonrelocatable, symbols,  
   List, 2-135  
 .NOOPERATOR Command (BATCH), 3-4, 3-44  
 /NORESTART Switch, 3-31  
 /NOREWINDS Switch, 2-66  
 /NORMAL Switch, 2-66  
 /NOSCAN Switch, 2-248  
 /NOSEARCH Switch, 2-59, 2-96, 2-158  
 /NOSORT Switch, 2-66  
 /NOSUMMARY Switch, 2-67  
 /NOSUPPRESS Switch, 3-11,  
   3-13, 3-16, 3-18, 3-22  
   3-27, 3-34, 3-35, 3-39  
 /NOSYS Switch, 2-248  
 /NOTE Switch, 2-180, 2-187, 2-201, 2-287  
 /NOTITLE Switch, 2-67  
 /NOUNITS Switch, 2-67  
 /NOWRITE Switch, 2-248  
  
 Null  
   Extension, 2-19  
   Output record, 2-132  
   Saved segments, 2-132  
 /NULL Switch, 2-34, 2-180,  
   2-187, 2-201, 2-287  
 Number,  
   Charge, 2-48  
   Determine station, 2-296  
   Job terminal, 2-174  
   Logical station, 2-161  
   of blocks, 2-210, 2-129  
   Physical station, 2-161  
   Print station, 2-8  
   Program version, 2-1, 2-293  
   Project-Programmer, 2-48, 2-162, 3-36, 1-7,  
   3-28  
   Sequence, 3-37  
 Number sign, 1-12  
 Number to pool,  
   Return job, 2-146  
  
 /O Switch, 2-101, 2-119  
 Object programs,  
   Prepare, 2-5  
 Obtain job size, 1-5  
 Obtain job state, 1-5  
 Obtain program name, 1-5  
 Octal representation of  
   filename, 1-12  
 /OKBINARY Switch, 2-187,  
   2-202  
 /OKNONE SWITCH, 2-1  
 /OKNONE Switch, 2-67, 2-180,  
   2-188, 2-202, 2-287  
 /OLD Switch, 2-23, 2-59,  
   2-96, 2-159  
 OMOUNT Program, 2-72, 2-166  
 Open a file,  
   for creation, 2-37  
   for editing, 2-90, 2-280  
 .OPERATOR Command (BATCH),  
   3-4, 3-44  
 Operator control over  
   devices, 2-166  
 Operators,  
   Contents, 2-86  
 OPSER Command, 2-7  
 OPSER Program, 2-170  
 OPTION Command, 2-84  
 Option files, 1-14

- /OPTION Switch, 2-67
- ORADIX Command, 2-84
- Output.
  - BATCH, 3-6
  - Binary compare, 2-109
  - global symbols, 2-135
  - Job, 3-46
  - Log File, 3-46
  - Null record, 2-132
  - Source files, 2-153
  - Source compare, 2-109
  - Suppress, 1-4, 3-46
  - Terminate, 2-122
- OUTPUT Command, 2-84
- Output File (FUDGE2), 2-126
- Output queue.
  - Card punch, 2-31
  - Line printer, 2-184, 2-193
  - Plotter, 2-177
- Output spoolers, 3-1, 3-4
- /OUTPUT Switch, 2-202, 2-270
- Override system default switches, 1-14
  
- /P Switch, 2-102, 2-120, 2-135
- .PII Extension, 2-19
- Page limit.
  - Current physical, 2-242
  - Current virtual, 2-258
- /PAGE Switch, 2-203, 2-270
- /PAGES Switch, 3-31
- /PAPER Switch, 2-203
- Paper-tape punch queue, 2-193, 2-284
- Parameters.
  - Swapping, 2-48
- /PARITY Switch, 2-67
- PARTIAL ALLOCATION Message, 2-9
- Partial reinitialization, 2-281
- Pass devices, 2-214
- Passing LINK-10 switches, 1-25
- Passing LOADER Switches, 1-24
- SPASSWORD Card, 3-5, 3-28, 3-36
  
- Password masks, 1-7
- Passwords, 1-7, 2-162
- /PAUSE Switch, 2-72, 2-167
- Percent (%) character, 1-24
- Performance statistics.
  - Disk, 2-277
- Period (.), 1-3
- Period (.) in filename, 1-10
- Permanent switch, 1-18
- PJOB Command, 2-8, 2-174
- Place entries in entries, 2-4
- Place output in log file, 3-46
- PLEASE Command, 2-7, 2-175
- PLOT Command, 2-4, 2-177
- /PI.OT Switch, 2-180, 2-203, 3-22
- Plotter output queue, 2-177
- Plotter queue, 2-193
- .PLT, 2-194
- PLT:, 2-194
- PLTSPL, 3-1
- Point.
  - Alternate entry, 2-218
- Pointers.
  - TMPCOR, 2-48
- Pool.
  - Return device to monitor, 2-146
  - Return job number to, 2-146
- Pools.
  - Monitor device, 2-72
- Preparation commands.
  - Program, 2-5
  - Source file, 2-3
- Prepare object programs, 2-5
- PRESERVE Command, 2-4, 2-183
- Print
  - a file, 2-5, 2-290
  - available devices, 2-8, 2-221
  - file structures, 2-221
  - station number, 2-8
  - status information, 1-5
  - system information, 2-8
- PRINT Command, 2-4, 2-184
- /PRINT Switch, 2-188, 2-204, 3-22
- Printer.
  - Write to line, 2-4
- Printer output queue.
  - Line, 2-184, 2-193

Printer queue,  
   Line, 3-4  
 Priorities,  
   Job, 3-2, 3-8  
 Priority,  
   Job, 2-239  
   Standard, 2-239  
 /PRIORITY Switch, 2-34, 2-181,  
   2-188, 2-204, 2-271, 2-287, 3-31  
 Privilege bits, 2-48  
 Process errors, 3-43  
 Processing,  
   ASCII, 2-120  
   Binary, 2-120  
   Image, 2-120  
 Processing  
   Commands, 1-3  
   Jobs, 3-1  
   DECtape specifiers  
 Processor,  
   The standard, 1-19  
 Processor status, 2-48  
 Processor switch,  
   Binary, 1-20  
   List, 1-20  
   Source, 1-20  
 Processor switches, 1-19  
 Processor switches,  
   Interpreting, 1-20  
   List of, 1-20  
 Processor time limit,  
   Central, 2-250  
 Produce relocatable  
   binaries, 2-19  
 Produce source files, 2-5  
 Program,  
   ALCFIL, 2-9  
   COMPIL, 2-124  
   Compile a COBOL, 3-14  
   Compile a MACRO, 3-32  
   Compile a SNOBOL, 3-37  
   Compile an ALGOL, 3-10  
     2-6, 2-7, 2-26, 2-40  
   control commands, 2-5  
   DAEMON, 2-46, 2-79  
   DUMP, 2-81  
   examination command, 2-6  
   execution, 2-266  
   Execute, 2-92  
   FAILSAFE, 2-99  
   FILCOM, 2-106  
   File transfer, 2-119  
   FILEX, 2-119  
   FUDGE2, 2-126  
   GLOB, 2-134  
   GRIPE, 2-138  
   LINK-10, 1-24  
   Loader, 1-24  
   name, 2-48, 1-5  
   OMOUNT, 2-72, 2-166  
   OPSER, 2-170  
   preparation commands, 2-5  
   prepare object, 2-5  
   QUOLST, 2-210  
   REATA, 2-216  
   Restart a, 2-218  
   Save a, 2-225  
   SFTSRC, 2-246  
   Start a, 2-6, 2-7, 2-40  
     2-212, 2-223  
   Stop a user, 1-2  
   UMOUNT, 2-72, 2-166  
   version number, 2-1, 2-293  
 Program counter, 1-5  
 Program from cards,  
   Execute a, 3-16  
 Program name,  
   Obtain, 1-5  
   User, 2-48  
 Programmer numbers,  
   Wildcard, 3-28  
 Progress of job,  
   Determine, 1-5  
 Project-programmer numbers,  
   1-7, 2-48, 2-162, 3-36  
 Properties,  
   Terminal, 2-252  
 PROTECT Command, 2-4, 2-191  
 Protect files, 2-146  
 /PROTECT Switch, 2-34  
   2-181, 2-188, 2-204,  
   2-271, 2-287  
 Protection,  
   standard, 2-183, 2-191  
 Protection codes,  
   Access, 1-10, 2-191  
   Redefining, 1-10  
 /PROTECTION Switch, 2-67  
 Pseudo-TTYs, 2-170  
 .PTP, 2-194  
 PTP:, 2-194, 2-284  
 PTPSPL, 3-1  
 PUNCH Command, 2-4, 2-284  
 Punch output queue,  
   Card, 2-31

- Punch queue,
  - Card, 2-193
  - Paper-tape, 2-193, 2-284
- /PUNCH Switch, 2-34, 2-204, 3-22
  
- /Q Switch, 2-102, 2-107, 2-108, 2-120, 2-135
- QMANGR, 3-1, 3-2
- Question mark as wild character, 1-11
- Queue,
  - BATCH Input, 1-6, 2-193, 2-194, 2-267
  - Card punch, 2-193
  - Card punch output, 2-31
  - High-priority scheduler run, 2-240
  - Line printer, 3-4
  - Line printer output, 2-184, 2-193
  - Maintain system, 3-2
  - Manager, 3-1, 3-2
  - Paper-tape punch, 2-193, 2-284
  - Plotter, 2-193
  - Plotter output, 2-177
- QUEUE Command, 2-4, 2-193
- :QUEUE Command (OPSER), 2-172
- QUOLST Command, 2-8
- QUOLST Program, 2-210
- Quota,
  - List is, 2-166
  - Logged-input, 2-8
- QUOTA.SYS, 2-166
  
- R Command, 2-6, 2-212
- R Command (SETSRC), 2-247
- /R Switch, 2-102, 2-120, 2-135
- Read binary files, 2-134
- Reading of control file,
  - Interrupt sequential, 3-41
- Reads,
  - Disk blocks, 2-48
  - Incremental disk, 1-5
- Ready character, 1-3
- Real-time status word, 2-48
- Reallocate space, 2-9
- REASSIGN Command, 2-2-214
- REATTA Command, 2-7
- REATTA Program, 2-216
- Record,
  - Output null, 2-132
  - Trailer, 2-100
- Recovery aid,
  - Error, 3-41
- Recovery routines,
  - Error, 3-40
- Redefining protection codes, 1-10
- Reel identification, 2-167
- REELID, 2-260
- REENTER Command, 2-6, 2-218
- Reinitialization,
  - Partial, 2-218
- .REL files,
  - Load, 2-92
- /REL Switch, 2-59, 2-96, 2-159
- Relative date argument, 1-12
- Relative time argument, 1-12
- Relocatable binaries,
  - Produce, 2-19
- Relocatable binary modules, 2-126
- \$RELOCATABLE Card, 3-9, 3-36
- Relocatable symbols,
  - List, 2-135
- Remove a file structure, 2-72
- /REMOVE Switch, 2-35, 2-73, 2-181-2-188, 2-205, 2-288
- Rename a file, 2-4, 2-183
- RENAME Command, 2-4, 2-219
- /REPORT Switch, 2-188, 2-205
- Reporting,
  - Media error, 2-166
  - SPRINT-10 Error, 3-48
- Representation of filename,
  - Octal, 1-12
- Requeue a job, 3-45
- .REQUEUE Command, 3-45
- Rescanning input lines, 2-26
- RESOURCES Command, 2-8, 2-221
- :RESOURCES Command (OPSER), 2-172
- Response counter, 2-48
- Responsibilities,
  - SPRINT-10, 3-1
- Restart a program, 2-218
- /RESTART Switch, 2-206, 2-271
- Restore disk files, 2-4
- Restore files, 2-100, 2-101, 2-102
- :RESTRICT Command (OPSER), 2-172
- Restricted devices, 2-2, 2-214

- Restrictions,
  - \$DATA Card, 3-20
- Return allocated core, 2-146
- Return device to monitor pool, 2-146
- Return devices to device pool, 2-54
- Return job number to pool, 2-146
- RETURN Key, 1-6
- Retype input line, 1-5
- .REVIVE Command, 3-46
- :REVIVE Command (OPSER), 2-172
- Rewind a DECTape, 2-222, 2-291
- Rewind a magnetic tape, 2-4, 2-222, 2-291
- Rewind a tape, 2-4, 2-129, 2-222, 2-291
- REWIND Command, 2-4, 2-222
- /REWINDS Switch, 2-67
- RIGHTMARGIN Command, 2-84
- /RONLY Switch, 2-167
- Routines,
  - Error recovery, 3-40
- RUBOUT Key, 1-4
- RUN Command, 2-6, 2-85, 2-223
- Run LINED, 2-37
- /RUN Switch, 2-67
- Run time, 2-282
- Run time,
  - Incremental, 1-5
  - Job, 2-8
- Running SPRINT-10, 3-7
- /RUNOFFSET Switch, 2-67
  
- /S Switch, 2-102, 2-108, 2-135
- Sample card deck, 1-3
- Sample jobs,
  - Batch, 3-49
- Save a program, 2-225
- SAVE Command, 2-5, 2-225, D-1
- Save disk files, 2-4, 2-99, 2-102
- Save files, 2-100, 2-101, 2-146
- Saved null segments, 2-132
- Saves,
  - Multiple, 2-101
- SBLK, 2-120
- /SCAN Switch, 2-248
- SCHED Command, 2-8, 2-227
- Schedule bits,
  - Type out, 2-227
- Schedule jobs, 3-2, 3-8
  
- Scheduler run queue,
  - High-priority, 2-240
- Search files, 2-134
- Search list,
  - Alter a, 2-246
  - Job, 2-8, 2-246
  - Manipulate, 2-246
  - System, 2-8, 2-246
- /SEARCH Switch, 2-59, 2-96, 2-159, 3-28
- Segment,
  - Sharable high, 2-264
  - Saved null, 2-132
- /SELF Switch, 2-23, 2-59, 2-96, 2-159
- Semicolon (:), 1-6, 3-9
- SEND Command, 2-7, 2-228
- :SEND Command (OPSER), 2-172
- Sending messages, 2-7
- /SEQ Switch, 2-35, 2-181, 2-188, 2-206, 2-271, 2-288
- \$SEQUENCE Card, 3-37
- Sequence number, 3-37
- /SEQUENCE Switch, 3-31
- Set access date word, 2-100
- SET BLOCKSIZE Command, 2-3, 2-230
- SET BREAK Command, 2-231
- SET CDR Command, 2-3, 2-234
- :SET Command (OPSER), 2-172
- SET CPU Command, 2-3, 2-235
- Set creation date, 2-100
- SET DENSITY Command, 2-3, 2-237
- SET DSKFUL Command, 2-6, 2-238
- SET DSKPRI Command, 2-3, 2-239
- Set filename for CDR, 2-234
- SET HPQ Command, 2-3, 2-240
- SET PHYSICAL GUIDELINE Command, 2-242
- SET PHYSICAL LIMIT Command, 2-242
- SET SPOOL Command, 2-3, 2-244
- Set tape density, 2-100, 2-103
- SET TIME Command, 2-8, 2-250
- Set time word, 2-100
- SET TTY Command, 2-3, 2-252
- SET VIRTUAL LIMIT Command, 2-258
- SET WATCH Command, 2-8, 2-259
- SETSRC Command, 2-8
- SETSRC Program, 2-246
- Sharable high segment, 2-264
- Sign,
  - Number, 1-12
- Sign, (\$),
  - Dollar, 1-6

- Sign (=),
  - Equal, 1-2
- .SILENCE Command, 3-46
- :SILENCE Command (OPSER), 2-172
- Simple block format, 2-120
- /SINCE Switch, 2-35, 2-67, 2-181, 2-189, 2-206, 2-288
- /SINGLE Switch, 2-167
- Size,
  - Cluster, 2-9
  - Obtain, 1-5
- SKIP Command, 2-4, 2-262
- Slash, (BATCH), 3-7
- :SLOGAN Command (OPSER), 2-172
- /SLOW Switch, 2-68
- .SNO Extension, 2-19
- \$\$NOBOL Card, 3-37
- SNOBOL Compiler, 2-19
- SNOBOL Program,
  - Compile a, 3-37
- /SNOBOL Switch, 2-23, 2-59, 2-97, 2-159
- /SORT Switch, 2-68
- Source compare, 2-107, 2-109
- Source files,
  - List, 2-153
  - Output, 2-153
  - Processor Switch, 1-20
  - Preparation commands, 2-3
  - Produce, 2-5
  - Translate, 2-155, 2-92
  - Translates, 2-55
- Space,
  - Allocate, 2-3
  - Free system, 2-210
  - Reallocate, 2-9
- Space,
  - a tape backwards, 2-15, 2-3
- Space for new file,
  - Allocate, 2-9
- Space used,
  - Disk, 2-210
- Spaces in commands, 1-6
- /SPACING Switch, 2-189, 2-206
- Special character,
  - Control-O, 1-4
- Special Character,
  - Control-R, 1-5
- Special Character,
  - Control-T, 1-5
  - Control-U, 1-4
- Special characters, 1-2, 1-4
- Specification,
  - Change file, 2-4, 2-219
  - Control file, 2-194
  - File, 1-10
  - Input, 2-194
  - Listing file, 2-194
  - Log file, 2-194
- Specification in LINK-10, Switch, 1-25
- Specifiers,
  - DECtape Format, 2-119
  - DECtape Processing, 2-120
  - File format, 2-120
- SPOOL Command,
  - SET, 2-3, 2-244
- Spooler,
  - Card-reader, 2-234
  - Input, 1-5, 3-1
  - Input mode, 1-2
  - Output, 3-1, 3-4
- Spooling control bits, 2-48
- Spooling devices, 2-244
- SPRINT-10, 1-2, 1-5, 3-1
- SPRINT-10,
  - Running, 3-7
- SPRINT-10 Control Cards, 3-9
- SPRINT-10 Error Reporting, 3-48
- SPRINT-10 Files, 3-1
- SPRINT-10 Messages, 3-47
- SPRINT-10 Responsibilities, 3-1
- Square brackets in access codes, 1-10
- SSAVE Command, 2-5, 2-264, D-1
- Stacker,
  - Control commands to, 1-6
- Standard priority, 2-239
- Standard processor, 1-19
- Standard protection, 2-183, 2-191
- Standard translator, 2-19
- Start a program, 2-6, 2-7, 2-40, 2-212, 2-223
- Start after a HALT, 2-26
- Start after Control-C, 2-26
- START Command, 2-6, 2-266
- /START Switch, 2-35, 2-189, 2-206, 2-271, 2-288
- State,
  - Job, 2-292
  - Obtain job, 1-5
- Station,
  - Change logical, 2-161
- Station message,
  - Central, 2-228
- Station number,
  - Determine, 2-296
  - Logical, 2-161



Physical, 2-161  
 Print, 2-8  
 Statistics,  
   Disk performance, 2-277  
   Job, 2-8, 2-259  
 Status,  
   Processor, 2-48  
   System, 2-273  
 Status information, 2-273, 2-292  
 Status information,  
   Job, 2-8  
   Print, 1-5  
 Status word,  
   Job, 2-48  
   Real-time, 2-48  
 STDAT Message Identifier, 3-47  
 STERR Message Identifier, 3-47  
 STMSG Message Identifier, 3-47  
 Stop,  
   a device, 2-146  
   a job, 2-6, 2-139  
   a user program, 1-2  
 :STOP Command (OPSER), 2-172  
 STOPR Message Identifier, 3-47  
 /STRS Switch, 2-35, 2-181, 2-289,  
   2-206, 2-288  
 Structure,  
   Access a file, 2-166  
   Remove a file, 2-72  
 Structure name,  
   Device, 2-48  
   File, 2-48, 1-10  
 Structures,  
   Print file, 2-221  
 STSUM Message Identifier, 3-47  
 Sub-file directory, 1-10  
 Subjobs, 2-170  
 SUBMIT Command, 2-4, 2-267  
 Submitting a Job with a Control file, 3-8  
 Submitting a Job with Files (BATCH), 3-6  
 Submitting jobs to BATCH, 3-4  
 /SUMMARY Switch, 2-68  
 SUPERSEDE Command, 2-85  
 Suppress all output from job, 3-46  
 Suppress output, 1-4  
 /SUPPRESS Switch, 3-11, 3-13, 3-16,  
   3-19, 3-22, 3-27, 3-34, 3-35, 3-39  
 Swapping parameters, 2-48  
 Switch,  
   /nL, 2-107, 2-108  
   /nU, 2-107  
   /026, 3-12, 3-14, 3-16, 3-19,  
     3-23, 3-27, 3-34, 3-35, 3-39  
   /2, 2-103  
   /5, 2-103, 2-120  
   /6, 2-77  
   /8, 2-103  
   /A, 2-99, 2-107, 2-120, 2-135  
   /ACCESS, 2-65  
   /AFTER, 2-32, 2-178, 2-185, 2-196,  
     2-268, 2-285, 3-29  
   /ALGOL, 2-20, 2-56, 2-92, 2-155  
   /ALLOC, 2-65  
   /ASCII, 3-11, 3-13, 3-15, 3-17,  
     3-20, 3-33, 3-35, 3-38  
   /B, 2-99, 2-107, 2-120, 2-129  
   /BEFORE, 2-32, 2-65, 2-178,  
     2-185, 2-196, 2-285  
   /BEGIN, 2-32, 2-178,  
     2-185, 2-196, 2-285  
   /BIN, 2-20, 2-56, 2-92, 2-155  
   /BINARY, 3-17, 3-21  
   Binary processor, 1-20  
   /BLISS, 2-56, 2-93, 2-20  
   /BLOCKS, 2-65  
   /C, 2-76, 2-99, 2-107, 2-120  
   /CARDS, 2-196, 2-268, 3-29  
   /CHARGE, 2-196  
   /CHECK, 2-72, 2-166, 2-167  
   /CHECKSUM, 2-65  
   /COBOL, 2-20, 2-56, 2-93, 2-156  
   /COMPILE, 2-20, 2-56, 2-93, 2-156  
   /COPIES, 2-32, 2-178,  
     2-185, 2-197, 2-285  
   /CORE, 2-197, 2-268, 3-29  
   /CPUNCH, 3-21  
   /CREATE, 2-33, 2-178, 2-185,  
     2-197, 2-247, 2-268, 2-286  
   /REF, 2-21, 2-56, 2-93,  
     2-156, 3-27, 3-33  
   /D, 2-100, 2-120  
   /DDT, 2-56  
   /DEADLINE, 2-33, 2-179, 2-186,  
     2-197, 2-269, 2-286, 3-29  
   /DEFER, 2-198  
   /DENSITY, 2-65  
   /DEPEND, 2-198, 2-269, 3-30  
   /DETAIL, 2-65  
   /DISPOSE, 2-33, 2-179, 2-186,  
     2-198, 2-269, 2-286  
   /E, 2-100, 2-120, 2-135  
   /EOTS, 2-66  
   /ERROR, 3-30  
   /F, 2-33, 2-100, 2-119,  
     2-135, 2-179, 2-186,  
     2-199, 2-269, 2-286  
   /F10, 2-21, 2-57, 2-93, 2-157  
   /F40, 2-22, 2-57, 2-93, 2-157

/FAST, 2-66  
 /FEET, 2-199, 2-269, 3-31  
 /FILE, 2-186, 2-199, 2-286  
 /FILES, 2-66  
 /FORMS, 2-33, 2-187, 2-200  
 /FOROTS, 2-56, 2-93, 2-156  
 /FORSE, 2-56, 2-93, 2-156  
 /FORTRAN, 2-21, 2-56, 2-93, 2-156  
 /FUDGE, 2-21, 2-57, 2-93,  
     2-124, 2-156  
 /G, 2-76, 2-100, 2-120  
 /H, 2-76, 2-100, 2-107,  
     2-108, 2-119, 2-135  
 /HEADER, 2-187, 2-200  
 /HELP, 2-66, 2-72, 2-167, 2-200  
 /HOLLERITH, 3-31  
 /I, 2-100, 2-120  
 /IMAGE, 3-18  
 /J, 2-100  
 /K, 2-101, 2-129  
 /KA10, 2-22, 2-57, 2-93, 2-157  
 /KI10, 2-57, 2-93, 2-157  
 /KILL, 2-33, 2-179, 2-187,  
     2-200, 2-270, 2-286  
 /L, 2-76, 2-101, 2-120, 2-135  
 /LIBRARY, 2-57, 2-93, 2-157  
 /LIMIT, 2-34, 2-179, 2-187,  
     2-201, 2-287  
 /LINK, 2-57, 2-93, 2-157  
 /LIST, 2-22, 2-34, 2-58,  
     2-66, 2-93, 2-157, 2-180,  
     2-187, 2-201, 2-270, 2-287  
 List processor, 1-20  
 /LMAP, 2-58, 2-95, 2-157  
 /LOADER, 2-58, 2-95, 2-157  
 /LOCATE, 3-31  
 /LOG, 2-187, 2-201  
 /M, 2-101, 2-119, 2-135  
 /MACRO, 2-22, 2-58, 2-95, 2-157  
 /MACX11, 2-58, 2-95, 2-158  
 /MANTIS, 2-22, 2-58, 2-95, 2-158  
 /MAP, 2-58, 2-95, 2-158, 3-18  
 /MARKS, 2-66  
 /MODIFY, 2-34, 2-180, 2-187  
     2-201, 2-270, 2-287  
 /MULTI, 2-167  
 /N, 2-76, 2-101, 2-135  
 /NAME, 3-31  
 /NEW, 2-22, 2-34, 2-58, 2-95,  
     2-158, 2-180, 2-187, 2-201,  
     2-248, 2-270, 2-287  
 /NOBIN, 2-23, 2-58, 2-96, 2-158  
 /NOCHECKSUM, 2-66  
 /NOCOMPILE, 2-23, 2-59, 2-96, 2-158  
 /NOCREATE, 2-247  
 /NODETAIL, 2-66  
 /NOEOTS, 2-66  
 /NOLIB, 2-248  
 /NOLIST, 2-23, 2-59, 2-96, 2-158,  
     3-11, 3-13, 3-27, 3-33  
 /NOMANTIS, 2-23, 2-59, 2-96, 2-158  
 /NOMARKS, 2-66  
 /NONEW, 2-248  
 /NORESTART, 3-31  
 /NOREWINDS, 2-66  
 /NORMAL, 2-66  
 /NOSCAN, 2-248  
 /NOSEARCH, 2-59, 2-96, 2-158  
 /NOSORT, 2-66  
 /NOSUMMARY, 2-67  
 /NOSUPPRESS, 3-11, 3-13, 3-16,  
     3-18, 3-22, 3-27, 3-34,  
     3-35, 3-39  
 /NOSYS, 2-248  
 /NOTE, 2-180, 2-187, 2-201, 2-287  
 /NOTITLE, 2-67  
 /NOUNITS, 2-67  
 /NOWRITE, 2-248  
 /NULL, 2-34, 2-180, 2-187,  
     2-201, 2-287  
 /O, 2-101, 2-119  
 /OKBINARY, 2-187, 2-202  
 /OKNONE, 2-1, 2-67, 2-180, 2-188,  
     2-202, 2-287  
 /OLD, 2-23, 2-59, 2-96, 2-159  
 /OPTION, 2-67  
 /OUTPUT, 2-202, 2-270  
 /P, 2-102, 2-120, 2-135  
 /PAGE, 2-203, 2-270  
 /PAGES, 3-31  
 /PAPER, 2-203  
 /PARITY, 2-67  
 /PAUSE, 2-72, 2-167  
 /Permanent, 1-18  
 /PHYSICAL, 2-34, 2-67, 2-180, 2-188,  
     2-203, 2-270, 2-287  
 /PLOT, 2-180, 2-203, 3-22  
 /PRINT, 2-188, 2-204, 3-22  
 /PRIORITY, 2-34, 2-181, 2-188,  
     2-204, 2-271, 2-287, 3-31  
 /PROTECT, 2-34, 2-181, 2-188,  
     2-204, 2-271, 2-287  
 /PROTECTION, 2-67  
 /PUNCH, 2-34, 2-204, 3-22  
 /Q, 2-102, 2-107, 2-108,  
     2-120, 2-135  
 /R, 2-102, 2-120, 2-135  
 /REL, 2-59, 2-96, 2-159

/REMOVE, 2-35, 2-73, 2-181,  
     2-188, 2-205, 2-288  
 /REPORT, 2-188, 2-205  
 /RESTART, 2-106, 2-271  
 /REWINDS, 2-67  
 /RONLY, 2-167  
 /RUN, 2-67  
 /RUNOFFSET, 2-67  
 /S, 2-102, 2-108, 2-135  
 /SCAN, 2-248  
 /SEARCH, 2-59, 2-96, 2-159, 3-28  
 /SELF, 2-23, 2-59, 2-96, 2-159  
 /SEQ, 2-35, 2-181, 2-188,  
     2-206, 2-271, 2-288  
 /SEQUENCE, 3-31  
 /SINCE, 2-35, 2-67, 2-181,  
     2-189, 2-206, 2-288  
 /SINGLE, 2-167  
 /SLOW, 2-68  
 /SNOBOL, 2-23, 2-59, 2-97, 2-159  
 /SORT, 2-68  
 /Source processor, 1-20  
 /SPACING, 2-189, 2-206  
 /START, 2-35, 2-189, 2-206,  
     2-271, 2-288  
 /STRS, 2-35, 2-181, 2-189,  
     2-206, 2-288  
 /SUMMARY, 2-68  
 /SUPPRESS, 3-11, 3-13, 3-16, 3-19,  
     3-22, 3-27, 3-34, 3-35, 3-39  
 /SYS, 2-23, 2-59, 2-97,  
     2-159, 2-248  
 /T, 2-76, 2-102, 2-119, 2-129  
 /TAG, 2-207, 2-271  
 /TAPE, 2-207, 2-288  
 /Temporary, 1-18  
 /TIME, 2-207, 2-271, 3-32  
 /TITLES, 2-68  
 /TPLOT, 2-208, 2-271, 3-32  
 /TPUNCH, 3-22  
 /U, 2-102, 2-108  
 /UNPRESERVED, 2-288  
 /UNIQUE, 2-208, 2-271, 3-32  
 /UNITS, 2-68  
 /UNPRESERVED, 2-35, 2-181, 2-189, 2-208  
 /V, 2-76, 2-103, 2-119  
 /VID, 2-167  
 /W, 2-103, 2-107, 2-129  
 /WENABL, 2-168  
 /WIDTH, 2-68, 3-11, 3-14, 3-16,  
     3-19, 3-22, 3-27, 3-34  
     3-35, 3-39  
 /WLOCK, 2-168  
 /WORDS, 2-68  
 /WRITE, 2-248  
 /WRITTEN, 2-68  
 /X, 2-103, 2-107, 2-136  
 /Y, 2-103  
 /Z, 2-77, 2-103, 2-120, 2-129  
 /ZDEFER, 2-208  
 Switch specification in  
     LINK-10, 1-25  
 SWITCH.INI, 1-14  
 Switches,  
     Compile, 1-18, 2-20  
     Interpreting processor, 1-20  
     List of processor, 1-20  
     Override system default, 1-14  
     Passing LINK-10, 1-25  
     Passing LOADER, 1-24  
     Processor, 1-19  
 Switches to the  
     linking-loaders, 1-24  
 SYFILE Command, 2-85  
 Symbols,  
     Check for global, 2-134  
     DUMP, 2-86  
     List erroneous, 2-135  
     List fixed, 2-135  
     List nonrelocatable, 2-135  
     List of, 2-134  
     List relocatable, 2-135  
     Output global, 2-135  
 Symbols list,  
     Cross-referenced, 2-134  
 Symbols table,  
     Internal, 2-134  
 /SYS Switch, 2-23, 2-59, 2-97,  
     2-159, 2-248  
 SYS:ACT.SYS, 2-162  
 SYSTAT Command, 2-8, 2-273  
 :SYSTAT Command (OPSER), 2-172  
 System,  
     Accessing the, 1-7  
     Configuring the, 1-1  
     Gain access to, 2-162  
     Loading the, 1-1  
     Logging on the, 1-7  
 System default switches,  
     Override, 1-14  
 System facilities,  
     Allocating, 2-2  
 System information,  
     Print, 2-8  
 System information commands, 2-8  
 System jobs,  
     Communicating with, 1-1

- System queues,
  - Maintain, 3-2
- System search list, 2-8, 2-246
- System space,
  - Free, 2-210
- System status, 2-273
  
- T Command (SETSRC), 2-247
- /T Switch, 2-76, 2-102, 2-119, 2-129
- Table,
  - Configuration, 2-46
  - Feature, 2-46
  - Internal symbols, 2-134
- TABS in Commands, 1-6
- /TAG Switch, 2-207, 2-271
- Tape,
  - Advance a magnetic, 2-99, 2-129
  - Backspace a magnetic, 2-99, 2-129
  - Density for magnetic, 2-237
  - Directory of files on, 2-101
  - Move a magnetic, 2-262
  - Rewind a, 2-4, 2-129
  - Rewind a magnetic, 2-222, 2-291
  - Rewind magnetic, 2-103
  - Unload a magnetic, 2-291
- Tape backwards,
  - Move a, 2-3
  - Space a, 2-3
- Tape blocksize,
  - Magnetic, 2-230
- Tape density,
  - Set, 2-100, 2-103
- Tape error analysis, 2-166
- Tape forward,
  - Move a, 2-4
- /TAPE Switch, 2-207, 2-288
- Tape to disk transfer, 2-115
- Tape transfer,
  - Disk to, 2-115
- TAPEID, 2-121
- TDUMP Command, 2-85
- Technique,
  - type-ahead, 1-3
- TECO, 2-164
- TECO Command, 2-3, 2-280
- Temporary files, C-1
- Temporary switch, 1-18
- Terminal,
  - Attach to a job, 2-13
  - Characteristics, 2-252
  - Disconnect a, 2-7
  - Disconnect job's, 2-63
  - Free a, 2-63
  - Identification, 2-228
  - Initialization, 2-144
  - Job number, 2-174
  - Modes, 1-2
  - Properties 2-252
- Terminate,
  - a job, 3-24
  - card input, 3-23
  - input, 2-122
  - output, 2-122
  - a command line, 1-6
- Termination commands,
  - Job, 2-7
- Test,
  - Busy, 2-228
- Text Editor, 2-164
- Time,
  - Incremental day, 1-5
  - Incremental run, 1-5
  - Job run, 2-8
- Time argument,
  - Absolute, 1-12
  - Relative, 1-12
- Time arguments, 1-12
- TIME Command, 2-8, 2-146, 2-282
- TIME Command,
  - SET, 2-250, 2-8
- Time limit,
  - Central processor, 2-250
- ?TIME LIMIT EXCEEDED
  - Message, 3-44
- Time limit for job, 2-8
- Time limit in jiffies, 2-48
- Time of day,
  - Type the, 2-45
- /TIME Switch, 2-207, 2-271, 3-32
- Time used,
  - Total, 2-48
- Time word,
  - Set, 2-100
- TITLE Command, 2-85
- /TITLES Switch, 2-68
- :TLOG Command (OPSER), 2-173
- TMPCOR Pointers, 2-48
- Total time used, 2-48
- TP Command (SETSRC), 2-247
- /TPLOT Switch, 2-208, 2-271, 3-32
- TPUNCH Command, 2-4, 2-284
- /TPUNCH Switch, 3-22
- Trailer record, 2-100

Transaction File (FUDGE2), 2-126  
 Transfer,  
   a job, 2-7, 2-216  
   control, 3-42  
   Disk to tape, 2-115  
   File, 2-115, 2-4, 2-27  
   Program (FILE), 2-119  
   Tape to disk, 2-115  
 Translate a source file, 2-92  
 Translate source files, 2-155  
 Translates source files, 2-55  
 Translator,  
   Standard, 2-19  
 TS Command (SETSRC), 2-247  
 :TSILENCE Command (OPSER), 2-173  
 TTY Command,  
   SET, 2-252, 2-3  
 :TTYTST Command (OPSER), 2-173  
 Type a file, 2-290  
 Type amount of core, 2-29  
 TYPE Command, 2-5, 2-85, 2-290  
 Type current date, 2-8  
 Type out schedule bits, 2-227  
 Type the current date, 2-45  
 Type the time of day, 2-45  
 Type-ahead technique, 1-3  
  
 /U Switch, 2-102, 2-108  
 /nU Switch, 2-107  
 /UNPRESERVED Switch, 2-288  
 UFD Creation, 2-166  
 UMOUNT Program, 2-72, 2-166  
 Unique command names, 1-6  
 /UNIQUE Switch, 2-208, 2-271, 3-32  
 /UNITS Switch, 2-68  
 Unload a DECtape, 2-291  
 Unload a magnetic tape, 2-291  
 UNLOAD Command, 2-5, 2-291  
 /UNPRESERVED Switch, 2-35, 2-181,  
   2-189, 2-208  
 Unprotected files,  
   Delete, 2-146  
 :UNRESTRICT Command (OPSER), 2-173  
 Unrestricted devices, 2-2, 2-214  
 Update a file, 2-5, 2-126  
 Used,  
   Disk space, 2-210  
   Total time, 2-48  
 User control file, 3-2  
  
 User core,  
   Clear, 2-132, 2-212, 2-223  
 User data file, 3-1  
 User file directory, 1-10  
 User File Directory (UFD), 2-166  
 USER Message Indicator, 3-48  
 User mode, 1-2  
 User program,  
   Stop a, 1-2  
 User program name, 2-48  
 User's core area, 2-46  
 USESTAT Command, 1-5, 2-8, 2-292  
  
 /V Switch, 2-76, 2-103, 2-119  
 VERSION Command, 2-8, 2-293  
 Version number,  
   Program, 2-1, 2-293  
 Versions,  
   Compare file, 2-4, 2-106  
 /VID Switch, 2-167  
 VIRTUAL LIMIT Command,  
   SET, 2-258  
 Virtual page limit,  
   Current, 2-258  
  
 /W Switch, 2-103, 2-107, 2-129  
 WATCH bits, 2-48  
 WATCH Command,  
   SET, 2-259, 2-8  
 /WENABL Switch, 2-168  
 :WHAT Command (OPSER), 2-173  
 WHERE Command, 2-8, 2-296  
 WIDTH Command, 2-85  
 /WIDTH Switch, 2-68, 3-11, 3-14, 3-16,  
   3-19, 3-22, 3-27, 3-34, 3-35, 3-39  
 Wild character,  
   Asterisk as, 1-11  
   Question mark as, 1-11  
 Wildcard construction, 1-11  
 Wildcard programmer numbers, 3-28  
 /WLOCK Switch, 2-168  
 Word,  
   Job status, 2-48  
 /WORDS Switch, 2-68  
 Write a core image, 2-5, 2-79  
 Write a core image file, 2-46  
 Write a core image to  
   device, 2-225

Write an end-of-file mark, 2-91  
/WRITE Switch, 2-248  
Write to line printer, 2-4  
Writes,  
    Incremental disk, 1-5  
Written,  
    Disk blocks, 2-48  
/WRITTEN Switch, 2-68

/Y Switch, 2-103

/Z Switch, 2-77, 2-103, 2-120, 2-129  
/ZDEFER Switch, 2-208  
ZERO Command, 2-5, 2-297  
Zero-compression, D-1

/X Switch, 2-103, 2-107, 2-136  
XTRACT Command, 2-86

## HOW TO OBTAIN SOFTWARE INFORMATION

### SOFTWARE NEWSLETTERS, MAILING LIST

The Software Communications Group, located at corporate headquarters in Maynard, publishes newsletters and Software Performance Summaries (SPS) for the various Digital products. Newsletters are published monthly, and contain announcements of new and revised software, programming notes, software problems and solutions, and documentation corrections. Software Performance Summaries are a collection of existing problems and solutions for a given software system, and are published periodically. For information on the distribution of these documents and how to get on the software newsletter mailing list, write to:

Software Communications  
P. O. Box F  
Maynard, Massachusetts 01754

### SOFTWARE PROBLEMS

Questions or problems relating to Digital's software should be reported to a Software Support Specialist. A specialist is located in each Digital Sales Office in the United States. In Europe, software problem reporting centers are in the following cities.

Reading, England	Milan, Italy
Paris, France	Solna, Sweden
The Hague, Holland	Geneva, Switzerland
Tel Aviv, Israel	Munich, West Germany

Software Problem Report (SPR) forms are available from the specialists or from the Software Distribution Centers cited below.

### PROGRAMS AND MANUALS

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center.

Digital Equipment Corporation Software Distribution Center 146 Main Street Maynard, Massachusetts 01754	Digital Equipment Corporation Software Distribution Center 1400 Terra Bella Mountain View, California 94043
--	--

Outside of the United States, orders should be directed to the nearest Digital Field Sales Office or representative.

### USERS SOCIETY

DECUS, Digital Equipment Computer Users Society, maintains a user exchange center for user-written programs and technical application information. A catalog of existing programs is available. The society publishes a periodical, DECUSCOPE, and holds technical seminars in the United States, Canada, Europe, and Australia. For information on the society and membership application forms, write to:

DECUS Digital Equipment Corporation 146 Main Street Maynard, Massachusetts 01754	DECUS Digital Equipment, S.A. 81 Route de l'Aire 1211 Geneva 26 Switzerland
---	---





READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Report (SPR) form (see the HOW TO OBTAIN SOFTWARE INFORMATION page).

Did you find errors in this manual? If so, specify by page.

---

---

---

---

---

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

---

---

---

---

---

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

---

---

---

---

---

Please indicate the type of user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Non-programmer interested in computer concepts and capabilities

Name \_\_\_\_\_ Date \_\_\_\_\_

Organization \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_  
or  
Country

If you do not require a written reply, please check here.

-----**Fold Here**-----

-----**Do Not Tear - Fold Here and Staple**-----

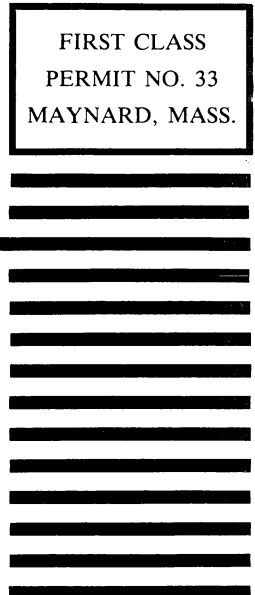
FIRST CLASS  
PERMIT NO. 33  
MAYNARD, MASS.

BUSINESS REPLY MAIL  
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

**digital**

Software Communications  
P. O. Box F  
Maynard, Massachusetts 01754





DIGITAL EQUIPMENT CORPORATION, Maynard, Massachusetts, Telephone: (617) 897-5111 • ARIZONA, Phoenix • CALIFORNIA, Sunnyvale, Santa Ana, Los Angeles, San Diego and San Francisco (Mountain View) • COLORADO, Engelwood • CONNECTICUT, Meriden • DISTRICT OF COLUMBIA, Washington (Riverdale, Md.) • FLORIDA, Orlando • GEORGIA, Atlanta • ILLINOIS, Northbrook • INDIANA, Indianapolis • LOUISIANA, Metairie • MARYLAND, Riverdale • MASSACHUSETTS, Cambridge and Waltham • MICHIGAN, Ann Arbor and Detroit (Southfield) • MINNESOTA, Minneapolis • MISSOURI, Kansas City and Maryland Heights • NEW JERSEY, Fairfield, Metuchen and Princeton • NEW MEXICO, Albuquerque • NEW YORK, Huntington Station, Manhattan, New York, Syracuse and Rochester • NORTH CAROLINA, Durham/Chapel Hill • OHIO, Cleveland, Dayton and Euclid • OKLAHOMA, Tulsa • OREGON, Portland • PENNSYLVANIA, Bluebell, Paoli and Pittsburgh • TENNESSEE, Knoxville • TEXAS, Dallas and Houston • UTAH, Salt Lake City • WASHINGTON, Bellevue • WISCONSIN, Milwaukee • ARGENTINA, Buenos Aires • AUSTRALIA, Adelaide, Brisbane, Crows Nest, Melbourne, Norwood, Perth and Sydney • AUSTRIA, Vienna • BELGIUM, Brussels • BRAZIL, Rio de Janeiro, Sao Paulo and Porto Alegre • CANADA, Alberta, Vancouver, British Columbia; Hamilton, Mississauga and Ottawa, Ontario; and Quebec • CHILE, Santiago • DENMARK, Copenhagen and Hellerup • FINLAND, Helsinki • FRANCE, Grenoble and Rungis • GERMANY, Cologne, Hannover, Frankfurt, Munich and Stuttgart • INDIA, Bombay • ISRAEL, Tel Aviv • ITALY, Milano • JAPAN, Osaka and Tokyo • MEXICO, Mexico City • NETHERLANDS, The Hague • NEW ZEALAND, Auckland • NORWAY, Oslo • PHILIPPINES, Manila • PUERTO RICO, Miramar and Santurce • REPUBLIC OF CHINA, Taiwan • SCOTLAND, West Lothian • SPAIN, Barcelona and Madrid • SWEDEN, Solna and Stockholm • SWITZERLAND, Geneva and Zurich • UNITED KINGDOM, Birmingham, Bristol, Edinburgh, London, Manchester, Reading and Warwickshire • VENEZUELA, Caracas