

# TOPS-10 Operating System Commands Manual

AA-0916D-TB

August 1980

This manual describes the monitor commands that are available to the TOPS-10 user.

This manual supersedes the *TOPS-10 Operating System Commands Manual*, order no. AA-0916C-TB.

**OPERATING SYSTEM:** TOPS-10 Version 7.01

**SOFTWARE:** GALAXY Version 2.0

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center. Outside the United States, orders should be directed to the nearest DIGITAL Field Sales Office or representative.

**NORTHEAST/MID-ATLANTIC REGION**

Technical Documentation Center  
Cotton Road  
Nashua, NH 03060  
Telephone: (800) 258-1710  
New Hampshire residents: (603) 884-6660

**CENTRAL REGION**

Technical Documentation Center  
1050 East Remington Road  
Schaumburg, Illinois 60195  
Telephone: (312) 640-5612

**WESTERN REGION**

Technical Documentation Center  
2525 Augustine Drive  
Santa Clara, California 95051  
Telephone: (408) 984-0200

**digital equipment corporation • marlboro, massachusetts**

First Printing, May 1971  
Revised, December 1971  
Revised, June 1972  
Revised, April 1973  
Revised, May 1974  
Revised, July 1975  
Revised, August 1977  
Revised, March 1978  
Revised, August 1980

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 document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

Copyright ©, 1971, 1980, Digital Equipment Corporation.  
All Rights Reserved.

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:

DEC	DECnet	IAS
DECUS	DECsystem-10	MASSBUS
Digital Logo	DECSYSTEM-20	PDT
PDP	DECwriter	RSTS
UNIBUS	DIBOL	RSX
VAX	EduSystem	VMS
		VT

## CONTENTS

	Page
PREFACE	ix
CHAPTER 1      INTRODUCTION	1-1
1.1      JOBS	1-1
1.2      PROJECT-PROGRAMMER NUMBERS AND PASSWORDS	1-1
1.3      OPERATING SYSTEM MODES	1-2
1.3.1      Interactive Mode	1-2
1.3.2      Batch Mode	1-2
1.4      INTERACTIVE LEVELS	1-2
1.4.1      Monitor Level	1-2
1.4.2      User Level	1-3
1.5      SPECIAL CHARACTERS	1-3
1.5.1      CTRL/C	1-4
1.5.2      DELETE Key	1-4
1.5.3      CTRL/W	1-4
1.5.4      CTRL/U	1-5
1.5.5      CTRL/R	1-5
1.5.6      CTRL/O	1-6
1.5.7      CTRL/S	1-6
1.5.8      CTRL/Q	1-6
1.5.9      CTRL/T	1-6
1.6      TYPE-AHEAD CAPABILITY	1-7
1.7      COMMAND FORMATS	1-7
1.7.1      Command Termination	1-8
1.7.2      Line Continuation	1-8
1.7.3      Command Arguments	1-8
1.7.3.1      Relative Date-Time Arguments	1-9
1.7.3.2      Absolute Date-Time Arguments	1-9
1.7.4      Command Switches	1-10
1.7.4.1      Temporary Switches	1-11
1.7.4.2      Permanent Switches	1-11
1.7.5      Comments	1-11
1.8      FILE SPECIFICATION	1-12
1.8.1      Device Names	1-13
1.8.1.1      Generic Device Names	1-14
1.8.1.2      Physical Device Names	1-15
1.8.1.3      File Structures	1-16
1.8.1.4      Logical Device Names	1-16
1.8.1.5      Ersatz Device Names	1-19
1.8.2      File Names	1-19
1.8.3      Directory Names	1-19
1.8.4      Protection Codes	1-20
1.8.5      File Daemon	1-22
1.9      SYSTEM DEFAULTS	1-22
1.10     WILDCARD CONSTRUCTIONS	1-22

CONTENTS (CONT.)

		Page
1.11	JOB SEARCH LIST	1-24
1.12	DIRECTORY PATHS	1-25
CHAPTER 2	SYSTEM COMMANDS AND PROGRAMS	2-1
2.1	FUNCTIONAL GROUPS OF COMMANDS	2-1
2.1.1	Job-Control Commands and Programs	2-1
2.1.2	Information Commands and Programs	2-2
2.1.3	Terminal-Control Commands and Programs	2-2
2.1.4	Terminal-Communication Commands	2-2
2.1.5	File-Handling Commands and Programs	2-2
2.1.6	Device-Handling Commands	2-3
2.1.7	Program-Preparation Commands and Programs	2-3
2.1.8	Program-Control Commands and Programs	2-3
2.1.9	Network Commands	2-3
2.1.10	MIC Commands	2-4
2.2	COMMAND DESCRIPTIONS	2-5
	ASSIGN Command	2-5
	ATTACH Command	2-7
	BACKSPACE Command	2-10
	BACKUP Program	2-11
	CCONTINUE Command	2-18
	CLOSE Command	2-21
	COMPILE Command	2-22
	CONTINUE Command	2-29
	COPY Command	2-31
	CORE Command	2-33
	CPUNCH Command	2-35
	CREATE Command	2-37
	CREDIR Program	2-38
	CREF Command	2-40
	CSTART Command	2-42
	D(eposit) Command	2-45
	DAYTIME Command	2-46
	DCORE Command	2-47
	DDT Command	2-53
	DEASSIGN Command	2-57
	DEBUG Command	2-59
	DELETE Command	2-67
	DETACH Command	2-69
	DIRECTORY Command	2-70
	DISABLE Command	2-79
	DISMOUNT Command	2-81
	DSK Command	2-86
	DUMP Command	2-88
	E(xamine) Command	2-89
	EDIT Command	2-90
	ENABLE Command	2-91
	EOF Command	2-92
	EXECUTE Command	2-93
	FILCOM Program	2-99
	FILE Command	2-108
	FINISH Command	2-113
	FUDGE Command	2-115
	GET Command	2-117
	GLOB Program	2-119
	HALT Command	2-123
	HELP Command	2-125



CONTENTS (CONT.)

	Page
2.2	
COMMAND DESCRIPTIONS (Cont.)	
INITIA Command	2-128
JCONTINUE Command	2-133
KJOB Command	2-134
LABEL Command	2-136
LIST Command	2-137
LOAD Command	2-138
LOCATE Command	2-144
LOGIN Command	2-146
MAKE Command	2-157
MERGE Command	2-158
MIC Commands	2-161
MOUNT Command	2-163
NETWORK Command	2-169
NODE Command	2-174
NSAVE Command	2-176
NSSAVE Command	2-176
OPSER Program	2-177
PJOB Command	2-182
PLEASE Command	2-183
PLOT Command	2-184
PRESERVE Command	2-186
PRINT Command	2-188
PROTECT Command	2-190
PUNCH Command	2-192
QUEUE Command	2-193
QUOLST Program	2-206
R Command	2-207
REASSIGN Command	2-209
REATTACH Program	2-211
REENTER Command	2-212
RENAME Command	2-213
RESOURCES Command	2-215
REWIND Command	2-216
RUN Command	2-217
SAVE Command	2-219
SSAVE Command	2-219
SCHED Command	2-221
SEND Command	2-223
SET BLOCKSIZE Command	2-225
SET BREAK Command	2-226
SET CDR Command	2-229
SET CPU Command	2-230
SET DEFAULT BUFFERS Command	2-232
SET DEFAULT PROTECTION Command	2-233
SET DENSITY Command	2-235
SET DSKFUL Command	2-236
SET DSKPRI Command	2-237
SET FLOATING POINT Command	2-238
SET HOST Command	2-239
SET HPQ Command	2-241
SET PHYSICAL Command	2-243
SET RETRY Command	2-245
SET SPOOL Command	2-247
SET TERMINAL or TERMINAL Command	2-249
SET TIME Command	2-250
SET TTY or TTY Command	2-252
SET VIRTUAL LIMIT Command	2-260

## CONTENTS (CONT.)

		Page
2.2	COMMAND DESCRIPTIONS (Cont.)	
	SET VIRTUAL LIMIT Command	2-260
	SET WATCH Command	2-262
	SKIP Command	2-270
	START Command	2-271
	SUBMIT Command	2-273
	SYSTAT Command	2-275
	TECO Command	2-281
	TIME Command	2-282
	TPUNCH Command	2-284
	TYPE Command	2-286
	UNLOAD Command	2-287
	USESTAT Command	2-288
	VERSION Command	2-289
	WHERE Command	2-292
	ZERO Command	2-294
CHAPTER 3	SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES	3-1
3.1	DIAGNOSTIC MESSAGES	3-1
3.2	ERROR MESSAGES	3-2
3.3	OTHER SYSTEM MESSAGES	3-2
3.4	FORMAT OF ERROR MESSAGES	3-2
3.5	SYSTEM MESSAGES	3-3
APPENDIX A	FUNCTIONAL GROUPS OF COMMANDS	A-1
APPENDIX B	SWITCH.INI FILES	B-1
APPENDIX C	COMPILE-CLASS COMMANDS	C-1
C.1	INDIRECT COMMANDS (@ CONSTRUCTION)	C-2
C.2	THE + CONSTRUCTION	C-3
C.3	THE = CONSTRUCTION	C-3
C.4	THE < > CONSTRUCTION	C-4
C.5	DEFAULT COMPILER	C-4
C.6	COMPILER SWITCHES	C-5
C.7	LINK SWITCHES	C-9
APPENDIX D	STANDARD SYSTEM NAMES	D-1
D.1	FILE NAME EXTENSIONS	D-1
D.2	RESERVED PROJECT-PROGRAMMER NUMBERS	D-6
D.3	ERSATZ DEVICE NAMES	D-7
APPENDIX E	CARD CODES	E-1
APPENDIX F	TEMPORARY FILES	F-1
APPENDIX G	SIXBIT/ASCII CHARACTER CODES	G-1
FIGURES		
FIGURE	2-1 DAEMON-Written File	2-48

CONTENTS (CONT.)

		Page	
TABLES			
TABLE	1-1	Device Names	1-14
	1-2	Generic Device Names	1-15
	1-3	Protection Codes for Field 1 Owner Protection Codes	1-20
	1-4	Protection Codes for Fields 2 and 3 Project-members and Other User Protections	1-21
	2-1	Job Information Data Words 5 through 52	2-49
	2-2	Terminal Hardware Characteristics	2-252
	2-3	Fill Characters	2-255
	A-1	Functional Groups of Commands/Programs	A-2
	C-1	Processor Switches	C-6
	D-1	File Name Extensions	D-1
	D-2	Project-Programmer Numbers	D-6
	D-3	Ersatz Device Names	D-7
	E-1	ASCII Card Codes	E-1
	E-2	DEC-026 Card Codes	E-4
	F-1	Temporary Files	F-1



## PREFACE

The TOPS-10 Operating System Commands Manual describes the commands available to the users of the TOPS-10 Operating System. See the TOPS-10 Operator's Guide for a description of all operator-privileged commands.

Commands to TOPS-10 are presented in alphabetical order in Chapter 2 for easy reference.

Reference material for assembly language programming can be found in the TOPS-10 Monitor Calls Manual, the MACRO Reference Manual, and the Hardware Reference Manual.

Getting Started with DECsystem-10 introduces first-time users to the TOPS-10 operating system command language. You should read this manual at least once before reading the TOPS-10 Operating System Commands Manual.

You should also be familiar with the following manuals:

Introduction to DECsystem-10 Software  
Getting Started With TOPS-10 Commands  
Introduction to Timesharing

### SYNOPSIS OF TOPS-10 OPERATING SYSTEM COMMANDS MANUAL

- Chapter 1 describes the components of the TOPS-10 operating system and how to use them.
- Chapter 2 lists the functional categories of commands, then lists all the commands alphabetically, and describes each command in detail. Examples accompany the command descriptions.
- Chapter 3 lists system error messages alphabetically and explains how to correct each error.
- The appendixes contain additional information and supplementary reference material.

## CONVENTIONS USED IN THIS MANUAL

The following conventions are used in this manual:

addr	Program or location address
arg	An argument to a command
c(addr)	The contents of an octal address
core	Referring to main (processor) memory
CTRL/x	A control character (Control characters are explained in Section 1.5.)
date	A date in the form of two digits for the day of the month, two digits for the month of the year, and two digits for the year (For example, 22-11-79 represents November 22, 1979.)
date-time	The date and time in the standard format. Refer to Section 1.7.3.
density	The density of a magnetic tape
dev: device-name devnnu devnn devu dev dev:	Any logical or physical device name (A colon (:) must be included when a device name is part of a file specification.)
[directory] [dir]	A directory name This can be either a UFD or an SFD.
expression	A numeric expression
file.ext	A file name and a file name extension, separated by a period
file name	A name of a file
file-spec	A file specification written in the format: dev:file.ext[directory]
fs	File structure
h	High (In reference to a job's high segment.)
hh:mm:ss	The time of day using a 24-hour clock (hh is hours, mm is minutes, and ss is seconds.)
id	An identifier
job	A job number
jobname	The name of the job.
l	Low (In reference to a job's low segment.)
letter	An alphabetic character
lh	The left half of an octal word

line-number	Referring to the number of the line
list	A list of arguments
logical-name log-name log	A logical device name, chosen by the user
memory	The decimal number of memory words measured in K or P (1K is equal to 1024 words; 1P (page) is equal to 512 words.)
MFD	A master file directory
<nnn>	A protection code
nnnn	A tape density
n	A decimal number, a unit number
^name^	A DECTape identifier
nodelist	A list of all the systems in a network environment
PPN	A project, programmer number
program	A program name
proj,prog	A project, programmer number
[proj,prog]	The name of a user-file directory
oo	An octal number
rh	The right half of an octal word
SFD	A sub-file directory
spec	A specification
str	A file structure name
/switch	A command line modifier
time	Referring to a time of day
UFD	A user-file directory
v	Version number
x	A numeric or text variable
\$	The symbol printed on your terminal when you press the ESCape (or ALTmode) key
^x	A character that is printed when you press the CONTROL key while you type a character key (For example, typing a CTRL/C prints ^C at your terminal.)

- DEL** Indicates when you should press the DELETE or RUBOUT key
- RET** Indicates when you should press the RETURN key
- ESC** Indicates when you should press the ESCape (or ALTmode) key

All examples, commands, switches, values, and arguments are shown in uppercase. This is for the sake of distinction. Examples are shown as they would appear on a terminal that does not have lowercase ability.

#### MANUALS REFERENCED

The following manuals are referred to in the text of this manual:

MANUAL	ORDER NUMBER
<u>Getting Started with DECSYSTEM-10</u>	AA-C790A-TB
<u>Getting Started with TOPS-10 Commands</u>	AA-C791A-TB
<u>Getting Started with Batch</u>	AA-D303A-TB
<u>TOPS-10/TOPS-20 Batch Reference Manual</u>	AA-H374A-TK
<u>TOPS-10 Monitor Calls Manual</u>	AA-0974C-TB
<u>LINK Reference Manual</u>	AA-0988C-TB
<u>MACRO Reference Manual</u>	AA-C780C-TB



## CHAPTER 1

### INTRODUCTION

TOPS-10 is the name of the Timesharing Operating System for use on DIGITAL's KI10, KL10 and KS10 systems. You communicate with the TOPS-10 operating system using the TOPS-10 command language. The TOPS-10 operating system is called the monitor.

#### 1.1 JOBS

TOPS-10 is a multiprogramming system; that is, the system transfers control rapidly among a number of jobs, so that all jobs appear to be running simultaneously. The term **job** refers to the entire sequence of steps that you start from your terminal or card deck. You start a job from your terminal by using the LOGIN command, and you use the KJOB command to end your job.

You can initiate a job at a central computer site or from a remote location.

After you initiate a job, you can initiate a second job without terminating the first. For example, you can initiate a timesharing job and, using the SUBMIT command, submit a second job for batch processing. (Refer to Chapter 2.)

When configuring and loading TOPS-10, the system administrator sets the maximum number of jobs the system can process.

#### 1.2 PROJECT-PROGRAMMER NUMBERS AND PASSWORDS

Only authorized users have access to TOPS-10. Your system administrator provides each authorized user with a project number, a programmer number, and a password. These numbers identify all users and their corresponding areas on file structures. When you specify a directory area, the project and programmer numbers are separated by a comma and are enclosed in square brackets (for example, [27,4072]).

The project numbers range from 1 to 377777 (octal). The programmer numbers range from 1 to 777777 (octal). Numbers 1 through 7 are reserved for DIGITAL, and numbers 400000 through 777777 are reserved for special purposes.

Your password is one to six characters long and is used when you log in to the system and when you attach to another job. (Refer to the ATTACH command, Chapter 2.) To maintain password secrecy, your password does not echo on the terminal. When you are using a terminal with local copy, a mask prints when you type your password, making the password unreadable.

## INTRODUCTION

### 1.3 OPERATING SYSTEM MODES

You can run jobs on the TOPS-10 operating system in two modes: interactive mode and batch mode.

#### 1.3.1 Interactive Mode

Jobs that run in interactive mode use a terminal to access the system. The language you use to run a job is the command language. The commands you type in from the terminal are received and processed by the command language interpreter of the TOPS-10 monitor. When your job is in interactive mode, you complete your work at two levels: monitor and user level. (See Section 1.4.)

#### 1.3.2 Batch Mode

Batch jobs communicate with the system in batch mode. They are input to the system from punched cards or from the terminal. Batch jobs are handled by the batch command interpreter. For information about TOPS-10 batch, refer to the following manuals:

- Getting Started With Batch
- TOPS-10/TOPS-20 Batch Reference Manual

### 1.4 INTERACTIVE LEVELS

When your terminal is in interactive mode, your job is at one of two levels, monitor level or user level.

#### 1.4.1 Monitor Level

When your job is at monitor level, you are communicating with the monitor. The command language interpreter processes each command you type. The monitor prompts you with a period (.). Every monitor command should follow this prompt. You end a monitor command by pressing the RETURN key (<RET>).

The format of each command is variable because many commands are followed by optional switches, arguments, or values. (See Section 1.7.)

Section 1.5 contains descriptions of the control characters and special keys you use when typing commands.

## INTRODUCTION

### 1.4.2 User Level

When you type the operating system command RUN, followed by a program name, your job moves from monitor level to user level. To move back to monitor level, type CTRL/C. If the program is not waiting for terminal input, it may be necessary to type CTRL/C twice. (Refer to Section 1.5.1.) Other commands also bring your job to user level. When your job is at user level, you are working with a program other than the operating system itself. Each program has its own set of commands and its own command interpreter. This manual describes some system programs. System programs usually prompt with an asterisk (\*).

At user level, control characters and special keys can have a different effect than they have at monitor level. Read the description of each program to determine the effects of the control characters and special keys.

The following example shows a dialogue with the system in an interactive job:

#### Example

Monitor level:

```
.R SETSRC (RET)
```

User level and program response:

```
*T (RET)  
DSKC: ,DSKB: ,FENCE
```

User level:

```
*A DSKN: (RET)
```

User level and program response:

```
*T (RET)  
DSKC: ,DSKB: ,DSKN: ,FENCE
```

CTRL/C to return to monitor level:

```
*^C
```

Monitor prompt at monitor level:

### 1.5 SPECIAL CHARACTERS

The command language interpreter recognizes several special characters and control characters that cause specific functions to be performed.

These special characters are described in the following sections.

## INTRODUCTION

### 1.5.1 CTRL/C

When you type a CTRL/C (control-C), the system interrupts your current running program and returns your terminal to monitor mode. You must type two CTRL/Cs if your program is not requesting input from your terminal (that is, if your program is executing). If you issue a CTRL/C while typing a command in monitor mode, this character causes the input line to be deleted. If you issue two CTRL/Cs while output is being printed on your terminal, the output is ended.

#### Example

This example shows a program prompt. CTRL/C brings your job to monitor level.

```
*^C
```

### 1.5.2 DELETE Key

When you press the DELETE (or RUBOUT) key, the monitor deletes the last character you typed. This function permits you to correct typing errors. When you press the DELETE key *n* times, the monitor deletes the last *n* characters that you typed. On hard-copy terminals the monitor echoes all deleted characters on your terminal and encloses the deleted characters in backslashes (\).

On video terminals the deleted characters that you typed are removed from the screen. However, you must inform the system of the kind of terminal you are using. (Refer to the SET TTY command in Chapter 2.) After you press RETURN or ESCape, you cannot use the DELETE function on that line.

#### Example

Type the characters SET TTY TYPE L036, press the DELETE key 3 times, then type the correct characters.

```
.SET TTY TYPE L036\630\A36 RET
```

### 1.5.3 CTRL/W

When you type CTRL/W (control-W), TOPS-10 deletes the last word you typed.

A word is defined as all spaces, tabs and alphanumeric characters until a nonalphanumeric character is encountered.

On video terminals, the deleted word is erased from the screen.

On hard-copy terminals, the deleted word is printed backwards between backslashes.

## INTRODUCTION

### Example

Type the characters SET TTY TYPE L036, type CTRL/W, and type the correct characters.

```
.SET TTY TYPE L036\630L\LA36 (RET)
```

### 1.5.4 CTRL/U

When you type CTRL/U (control-U), TOPS-10 deletes your current input line, back to the last time you pressed RETURN. On hard-copy terminals, the monitor responds with a carriage-return/line-feed, after which you can retype the line. On video terminals, the entire line that you typed is removed from the screen. After you have pressed RETURN, you can no longer use the line-editing features (for example, CTRL/U and DELETE) on that line.

### Example

Type CTRL/U.

```
.SETTTY ^U
```

Type the correct characters.

```
SET TTY LA36 (RET)
```

### 1.5.5 CTRL/R

When you type CTRL/R (control-R), the system reprints the current input line. If you type a line incorrectly, then make corrections using the DELETE key, the system will print the corrected line when you type CTRL/R. The following is an example of this operation using a hard-copy terminal:

### Example

If you type:

```
SET TTQ (DEL)Y NO (DEL)O FILE (DEL)L
```

The line appears as:

```
.SET TTQ\Q\Y NO\O\O FILE\E\L
```

If you then press CTRL/R:

```
.SET TTQ\Q\Y NO\O\O FILE\E\L ^R
```

The monitor prints:

```
SET TTY NO FILL
```

When you type CTRL/R, the system issues a carriage-return/line-feed before printing the corrected input line. The cursor or printing head of your terminal is left at its previous location.

## INTRODUCTION

### 1.5.6 CTRL/O

When you type CTRL/O (control-O), the monitor suppresses output to your terminal. This function is useful when a program begins to print a long message that you are not interested in reading. If you do not want to wait for the system to finish printing the message, you can stop the system from printing the message one of two ways. First, you can type two CTRL/Cs, but this action also stops the execution of the program. Second, you can type a single CTRL/O.

When you type CTRL/O, the system continues executing your program, but does not print any output on your terminal. The system begins printing to your terminal when one of the following conditions occurs:

- The executing program requests input from your terminal.
- The program ends and returns control to the monitor.
- You type CTRL/C twice, which returns control to the monitor.

To start the output to your terminal again, type another CTRL/O.

### 1.5.7 CTRL/S

When you type CTRL/S (control-S), the system stops output to your terminal. This control character works only after you have typed the SET TTY PAGE command. (See Chapter 2.) This control character is useful if you have a display terminal, and you have to stop the output to read it. To read the rest of the output, type CTRL/Q.

### 1.5.8 CTRL/Q

The CTRL/Q (control-Q) character resumes output to your terminal. CTRL/S and CTRL/Q are useful when you are using a display terminal. Used in conjunction with CTRL/S, you can stop and continue output to your terminal, thus reading a file before it scrolls off your display screen.

### 1.5.9 CTRL/T

When you type CTRL/T (control-T), the system prints status information pertaining to your job on your terminal. CTRL/T does not echo on your terminal. There are 8 items of information output to your terminal. These items are:

1. The incremental daytime (which is the time since you last issued a CTRL/T or a USESTAT command) or the time since you logged in if you have not issued a CTRL/T or a USESTAT command. (For example, DAY: :05:43.)
2. The incremental runtime, which is the CPU time used since you issued a CTRL/T, USESTAT command, LOGIN command, or TIME command. (For example, RUN:0.48.)
3. The incremental disk reads, which is the number of disk blocks read since you issued a CTRL/T, USESTAT command, LOGIN command, or DSK command. (For example, RD:75.)

## INTRODUCTION

4. Incremental disk writes, which is the number of disk blocks written since you issued a CTRL/T, USESTAT, LOGIN, or DSK command. (For example, WR:8.)
5. The program name. (For example, SOS.)
6. The memory size. (For example, 12+19P.)
7. The job state. The job state codes are described in the SYSTAT command description in Chapter 2. (For example, ^C.) An ampersand after the job state code indicates the job is locked in core. An asterisk indicates the job is being run or swapped.
8. The program counter, which is the address of the current instruction. (For example, PC:400275.)
9. The job state, which can be INPUT WAIT or OUTPUT WAIT. This item is printed only when you type CTRL/T from user level. (For example, OUTPUT WAIT FOR TTY21.)

This information can be obtained with the USESTAT command at monitor level. However, by typing CTRL/T, you can determine your job's progress without interrupting its execution. When you type CTRL/T, the character is not passed to your job as an input character. However, some programs activate a special interrupt feature when you type CTRL/T. (See the TOPS-10 Monitor Calls Manual.)

### Example

```
DAY: :05:43 RUN:0.48 RD:75 WR:8 SOS 12+19P ^C PS:400275
```

### 1.6 TYPE-AHEAD CAPABILITY

Type-ahead allows you to type another command without waiting for the system to respond to your first command. If you want two operations performed, you can begin typing the request for the second operation before you receive the prompt that the system prints after completing the first operation. If an error occurs during the first operation, the characters you typed ahead are ignored.

### 1.7 COMMAND FORMATS

The general format of each command is a line of ASCII uppercase or lowercase characters or a mixture of uppercase and lowercase characters. You type the commands after the monitor prints the TOPS-10 prompt (.). The following are examples of valid TOPS-10 commands:

```
.DIRECT  
.direct  
.dirECT
```

## INTRODUCTION

If you type spaces or tabs before a command name, the system ignores them. For example, the following commands produce the same result:

```
.DIRECT
.    DIRECT
```

The commands you type to the command language interpreter are one to six characters long. If you type any character past the sixth character, the system ignores it. You need only type enough characters to uniquely identify the command. However, when you are creating a batch control file, you should type the whole command, because uniqueness varies from monitor release to monitor release.

### 1.7.1 Command Termination

You terminate every command to the command language interpreter by pressing the RETURN key. For example:

```
.DIRECT (RET)
```

### 1.7.2 Line Continuation

You can continue a command line to some system programs (for example, DIRECT and QUEUE,) by placing a hyphen (-) as the last nonblank, noncomment character before the carriage-return/line-feed (that is, before you press the RETURN key). These programs treat continuation lines as part of the current command line. This feature allows you to type indefinitely long command lines. A line is terminated by a <RET> that is not preceded by a hyphen.

### 1.7.3 Command Arguments

You specify arguments to a command after the command name and separate them from the command name by a space. If the command language interpreter recognizes a command name, but cannot find a necessary argument, the monitor responds with the error message:

```
?TOO FEW ARGUMENTS
```

#### Example

The ASSIGN command requires arguments.

```
.ASSIGN (RET)
```

System error message:

```
?TOO FEW ARGUMENTS
```

TOPS-10 prompt:

After the system prints the error message, your terminal is left in monitor mode, as indicated by the monitor prompt. You can then retype the command.



## INTRODUCTION

1.7.3.1 **Relative Date-Time Arguments** - Certain commands require arguments that specify a date or time. Date and time arguments can be either relative or absolute. A relative argument specifies a certain length of time from the current date or time. The format of a relative argument is:

+number-of-daysD:hours:minutes:seconds

Where:

**number-of-days** is optional.

**D** is required if you specify number of days.

**hours** is optional if you specify number of days (with the letter D). Otherwise, hours is required.

**minutes** is optional.

**seconds** is optional. However, if you specify seconds, minutes must also be given or seconds will be interpreted as minutes.

You must type a colon to separate one field from the other. You can precede a relative argument with an optional plus (+) or minus (-) sign. The sign implies either past (-) or future (+). When you do not specify number of days, you must precede the time with a plus sign or a minus sign. For example:

-3D:4:27:21

means 3 days, 4 hours, 27 minutes, and 21 seconds ago. Similarly:

+4

means 4 hours from now.

1.7.3.2 **Absolute Date-Time Arguments** - An absolute argument specifies a particular date or time. The format of an absolute argument is:

weekday or date:hours:minutes:seconds

Where: **weekday** is the day of the week or one of the following: YESTERDAY, TODAY, TOMORROW.

This part of the argument is optional. The weekdays are abbreviated as follows:

SUNDAY	=	SUN
MONDAY	=	MON
TUESDAY	=	TUES
WEDNESDAY	=	WED
THURSDAY	=	THUR
FRIDAY	=	FRI
SATURDAY	=	SAT

## INTRODUCTION

**date** is optional and has one of the following formats:

day-month-year (21-OCT-79)

month-day-year (OCT-21-79)

numeric month-day-year (10-21-79)

The month can be abbreviated. (The abbreviations for the months are JA, F, MAR, AP, MAY, JUN, JUL, AU, S, O, N, and D).

The year and its preceding hyphen are optional, and, if given, can be one, two, or four digits. (For example, 0, 80, and 1980 will all be interpreted as the year 1980).

**hours** is based on a 24-hour clock and is required.

**minutes** is optional.

**seconds** is optional.

The following example specifies Wednesday at 9:15:06 AM.

### Example

```
WED:09:15:06
```

Because the date is known to be past or future from either the switch used (/BEFORE and /SINCE imply past, /AFTER implies the future) or by a plus or minus sign, an unspecified field is filled in so that the result is the next or last occurrence of the specified date. If you omit the time argument, the time defaults to 00:00:00 (midnight) if past, and 23:59:59 (11 o'clock, 59 minutes, and 59 second PM) if future.

### Examples

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

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

```
/SINCE:JUL-3-79 is since July 3, 1979 at midnight
```

### 1.7.4 Command Switches

You can modify some commands by including a switch in the command line. You precede each switch with a slash (/) and terminate it with a nonalphanumeric character, usually `(RET)`, a comma, or another switch. You can abbreviate the switch if its name remains unique. Abbreviation is not recommended for batch control files. Valid switches for each command are documented as part of the command descriptions in Chapter 2. The following is an example of a command, command argument, and command switch:

### Example

```
.PRINT MYFILE.EXT/COPIES:2
```

## INTRODUCTION

1.7.4.1 **Temporary Switches** - The switches for COMPILE-class commands are either temporary (local) or permanent (global). COMPILE-class commands are further described in Appendix C. A temporary switch applies only to the immediately preceding file. Do not place a space or comma between the file name and the switch. In the command construction:

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

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

1.7.4.2 **Permanent Switches** - A permanent switch (sometimes called a sticky switch) applies to all files following it on the command line, until you modify it by a subsequent switch. You separate the switch from the file name by spaces, commas, or a combination of both. For example, using the /MACRO switch:

### Examples

Temporary switch that affects PROG:

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

Temporary switch that affects PROG:

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

Permanent switch that affects TEST and SUBLET:

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

Permanent switch that affects TEST and SUBLET:

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

The COMPILE, LOAD, EXECUTE, and DEBUG command descriptions in Chapter 2 list the switches for these commands.

### 1.7.5 Comments

You can type a comment on the same line as a command by preceding the comment with a semicolon (;) or exclamation point (!). The monitor and the batch command language interpreters do not attempt to interpret the characters after the semicolon. Comment lines are useful when you are using a hard-copy terminal, or making control files for batch jobs. The following is an example of a line that contains both a command and a comment:

```
.DIRECT ;will list names of files in default area (REF)
```

## INTRODUCTION

### 1.8 FILE SPECIFICATION

The system stores programs and data as named files. When they are stored on DEctape or disk, files are identified by a file specification. The file specification includes the following identifications:

1. A device name or file structure name
2. A file name
3. A file name extension
4. An ordered list of directory names
5. An access protection code

The file specification is necessary to identify a disk file. If you issue a file specification for devices other than DEctape or disk, the system ignores them. File specifications are used to choose a file from a directory, a set of files belonging to a specific user. DEctapes and disks are the only directory-oriented devices. Items 4 and 5 in the above list do not apply to DEctapes.

The **device name** can be any valid device name described in the Section 1.5.2 of this manual. Always type a colon following the device name. Device names are discussed in Section 1.8.1. An example of a device name is DSKC:.

A **file name** is one to six alphanumeric characters. The system ignores all characters past the sixth. File names are discussed in Section 1.8.2. An example of a device name and a file name is DSKC:MYFILE.

The **file name extension** is a period (.) followed by zero to three characters. It is used to indicate the type of information in the file. (Refer to Appendix D for a list of standard file name extensions.) For the most efficient use of system resources, use only standard file name extensions, though other extensions can be valid. Most programs recognize file names and extensions consisting only of letters and digits. Often the term file name refers to both the file name and the file extension. An example of a device name, file name, and file extension is DSKC:MYFILE.TXT.

The **directory name** identifies the disk area where the file is stored. This list can be a user file directory (UFD) represented by the owner's project-programmer number, or a user file directory followed by one or more sub-file directories (SFDs). You must enclose a directory name in square brackets ([]). Directory names are discussed in Section 1.8.3. An example of a device name, file name, file extension and directory name is DSKC:MYFILE.TXT[21,589].

The **access protection code** of a file is a 3-digit octal code designating the users who can read or write the file. The code must be enclosed in angle brackets (< >), and you specify it only for output files. For a given file, users are divided into three groups: owner of the file, users with the same project number as the owner, and all other users. The standard protection code is <057>, allowing users in the owner's project to read and execute the file, and preventing access by all other users. The standard protection code may be different at your installation. Protection codes are described in Section 1.8.4. An example of a full file specification is DSKC:MYFILE.TXT[21,589]<055>.

## INTRODUCTION

The following information is necessary when you refer to a file:

- The file name.
- The device name, if the file is not on disk and not in your default search list.
- The directory name, if the file is not in your directory.

The following information is optional in a file specification:

- The file name extension.
- The device name, if the file is on a file structure in your search list.
- The directory name, if the file is in your directory.
- The protection code (if an output file).

### Examples

File name and file name extension:

TEXT.MAC

Physical device name and file name:

DTA3:FILEA

Generic device name, file name, file name extension and directory name:

DSK:PROG2.CBL[10,16]

A complete file specification: device name, file name, file name extension, directory name, and protection code:

DSKA:MAIN.F4[27,235]<057>

### 1.8.1 Device Names

TOPS-10 supports a number of input/output devices (peripherals) to handle data acquisition, storage, retrieval, and display. These devices are:

Card Punch	Card Reader	DECTape
Disk	Display Terminal	Hard-copy Terminal
Line Printer	Magnetic Tape	Paper-Tape Punch
Paper Tape Reader	Plotter	Graphics Display

The system allocates a device to your job when you request access to the device. (Refer to the ASSIGN and MOUNT commands in Chapter 2.) Until you request a device, it resides in the system pool of available resources.

## INTRODUCTION

To refer a device, you must specify a device name. In the command descriptions in this manual, places where you must supply a device name have the symbol `dev:` in the command line format. The types of device names are listed in Table 1.1 and are described in Sections 1.8.1.1 through 1.8.1.5.

Table 1-1  
Device Names

Type of Name	Meaning
generic	These specify a generic type of device such as a disk (DSK:) or a magnetic tape (MTA:).
physical	These specify a particular physical unit on a specific controller such as MTA1:, magnetic-tape unit number 1.
logical	These are substitute names for devices. You assign these names with the ASSIGN command.
ersatz	These are names for ersatz (pseudo-disk) devices normally used to contain libraries or special directories.
system defined logical	These are names defined by the system. Each name corresponds to a physical unit so long as the unit is declared to be the system default for that unit.

**1.8.1.1 Generic Device Names** - The most general type of device name is the generic device name. When you specify a generic name, the system selects a free unit of the device type specified. When your computer is in a network environment, the system chooses the device from those devices at your location, or, if none are available, the system chooses the device from the host (central) site.

A generic name can be two or three letters long and is followed by a colon (:). The generic names are listed in Table 1-2.

Two-character generic names are less specific than three-character generic names (for example, MT: means any magnetic tape unit, but MTA: means any magnetic tape unit on controller A). When you specify the generic name DSK: or DS:, the monitor uses your job search list to determine which disk device should be selected for you. (Refer to Section 1.11.)

## INTRODUCTION

Table 1-2  
Generic Device Names

DEVICE	3-Letter Device Name	2-Letter Device Name
Card punch	CDP:	CP:
Card reader	CDR:	CR:
Console terminal	CTY:	---
DEctape	DTx:	DT:
Disk	DSK:	DS:
Packs	DPx:	DP:
Fixed head	RPx:	RP:
	FHx:	FH:
	FSx:	FS:
Display	DIS:	---
Line printer	LPT:	LP:
lowercase/uppercase	---	LL:
uppercase	---	LU:
Magnetic tape	MTx:	---
7-track	---	M7:
9-track	---	M9:
Operator terminal	OPR:	---
Paper-tape punch	PTP:	PP:
Paper-tape reader	PTR:	PR:
Plotter	PLT:	---
Pseudo-terminal	PTY:	---
System library	SYS:	SY:
Terminal	TTY:	TT:

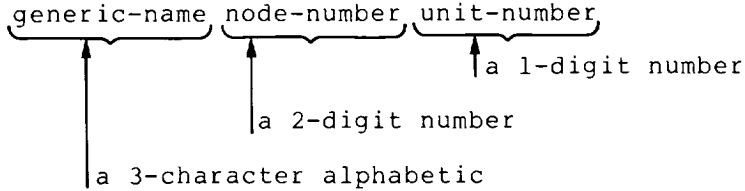
When you specify a generic name, the system selects the lowest-numbered available unit of the device type specified. There are two defaults when you specify a generic name:

1. When your program specifies DSK or DS, the system uses your job search list.
2. When you specify TTY: or TT:, the system always selects your terminal (assuming that these are not logical names).

**1.8.1.2 Physical Device Names** - Every I/O device has a physical name. This name consists of the generic name, a letter indicating the controller, and one numeric character indicating the unit number. For example, the generic name MTA: indicates any magnetic tape unit on controller A. However, MTAl: indicates magnetic tape unit 1 on controller A. Network node numbers can also be specified in a physical device name; formats for these types of device names are described in the following paragraph. Non-network nodes are assigned the node number 01.

## INTRODUCTION

You can choose a device according to its node system in a network. Device names of this kind are written in the following format:



### NOTE

You can assign a device from another node on the network, except when the other node is a host system. Refer to the description of the ASSIGN command in Chapter 2.

**1.8.1.3 File Structures** - The TOPS-10 operating system organizes disk devices into file structures. File structures are logical arrangements of 128-word blocks on one or more disk units. A file structure can exist on one disk unit, or it can be distributed over several disk units designated by a single name. However, two file structures cannot exist on the same unit.

The operator or system administrator assigns a name to every file structure when he or she defines the system's file structures. This name is one to five characters long, and cannot duplicate a physical device name, a unit name, or an existing file structure name. The recommended names for public file structures are DSKA:, DSKB:,...,DSKO: in order of decreasing speed.

**1.8.1.4 Logical Device Names** - You can assign a logical name to a physical device. The system will recognize the name that you assign, and associate the logical name with the physical device. You can assign a logical name to a device using the ASSIGN command.

The logical name that you assign may be up to six alphanumeric characters and may optionally be ended by a colon (:). Thereafter, the system will use the device that you associated with the logical name, when you or your programs specify that logical name. Logical names are cleared by the DEASSIGN command. That is, use the DEASSIGN command to disassociate logical names from devices. Logical names are also cleared when you log off the system, and when you specify another logical name for the same device.

Logical names are useful when you are writing a program that needs a device, but you will not know until program execution what device you will need. Use a logical name in the program to represent the device. Before you run the program use the ASSIGN command to associate the logical name with a physical device.

Logical names take precedence over physical names. Therefore, if you assign the logical name DSK: to the device MTA03: (magnetic tape unit 3 on controller A), all output to generic DSK will go to the magnetic tape.



## INTRODUCTION

Some names are recognized by the system as special system-defined logical names that the system translates into physical device names. One example is the generic device name OPR:. The generic name OPR: is the system-default logical name for the operator's terminal.

Therefore, the terminal specified as OPR: can change during system operation; but if you send a message to OPR:, the message will be routed to the last physical device declared to be the operator's terminal at your node.

All devices except disks and terminals can be designated as being **restricted devices**. You request a restricted device from the system pool of available resources by issuing the MOUNT command. This command requires operator intervention before the specified device is assigned to your job. The system deassigns a restricted device from your job when you log off the system or issue the DISMOUNT, DEASSIGN, or FINISH command.

Unrestricted devices are assigned to your job on a first-come, first-served basis when you issue the MOUNT or ASSIGN command. The device assignment remains in effect until you release the device by issuing a DEASSIGN command or a FINISH command, or you log off the system.

The following example illustrates the use of both **physical** and **logical device names**. In this example, the user is accessing a host computer in a network system with a node number of 26.

### Example

You request a DEctape drive with the logical name ABC:

```
.ASSIGN DTA: ABC:
```

The monitor gives you unrestricted device DTA260: (DEctape number 0 on controller A through host number 26). You then mount a DEctape on drive DTA260:

```
DTA260 ASSIGNED
```

You request any paper-tape punch and request that the logical name ABC: be assigned to it.

```
.ASSIGN PTP: ABC:
```

The monitor prints a warning message, telling you that the logical name was previously assigned to another device. The monitor changes the logical-name assignment, so that the logical name ABC: now refers to the paper-tape punch.

```
%LOGICAL NAME WAS IN USE, PTP260 ASSIGNED
```

You run the system program PIP.

```
.R PIP
```

You issue a command string to PIP asking that file FILEA.EXT be transferred from device DTA260: to logical device ABC: (physical device PTP260:).

```
*ABC:=DTA260:FILEA.EXT
```

## INTRODUCTION

You type a CTRL/C, returning your job to monitor mode.

\*^C

You request another DECTape drive and request that logical name DEF: be assigned to it.

.ASSIGN DTA: DEF:

The monitor prints a message telling you that all DECTape drives are in use by the specified jobs. The monitor does not assign a DECTape drive or a logical name to your job.

?ALREADY ASSIGNED TO JOB 13

You request that DECTape unit 0 at node 26 (which you already have assigned to you) be assigned the logical name DEF:

.ASSIGN DTA260: DEF:

DECTape unit 0 at node 26 takes on the logical name DEF:.

DTA260 ASSIGNED

You deassign the paper-tape punch, clearing the logical name ABC:.

.DEASSIGN PTP:

You run the system program PIP.

.R PIP

You request that the file FILEB is to be transferred from device DEF: to device ABC:

\*ABC:=DEF:FILEB

TOPS-10 prints an error message indicating that the logical device name ABC: is no longer assigned.

?DEVICE ABC DOES NOT EXIST

You type a CTRL/C, returning your job to monitor mode.

\*^C

You request drive DTA260:(DECTape drive 0 at node 26) again and request that the logical DEF: be changed to XYZ:

.ASSIGN DTA260: XYZ:

The system disassociates the logical name DEF: from DTA260: and clears the directory from memory.

DTA260 ASSIGNED

## INTRODUCTION

1.8.1.5 **Ersatz Device Names** - An ersatz device is a disk-simulated library. Although you specify an ersatz device like a file structure, an ersatz device represents a particular project-programmer number on a disk structure. Ersatz device names are three characters long. A complete list of ersatz device names that are used by DIGITAL is given in Appendix D.

### 1.8.2 File Names

Data and programs are stored in the system files. Files are used to arrange and protect data and programs. The essential part of a file specification is the file name. You name a file when you create, rename, or copy it.

The file name that you choose must be one to six alphanumeric characters long. It is possible to use non-alphanumeric characters, but some symbols have a special meaning to the system (see Section 1.10), and some programs may not recognize non-alphanumeric characters.

The file name is divided into the name of the file and the file extension. The format is:

FILE.EXT

Where FILE is the name you choose to distinguish the file, and .EXT is a standard or non-standard file extension. The period (.) is used to separate the file name from the file extension.

It is recommended that you use standard file extensions where applicable. The standard file name extensions are listed in Appendix D. File extensions are optional. You can name or specify a file without an extension.

File names and extensions can be changed using the RENAME command.

### 1.8.3 Directory Names

Your directory is a file that serves as an index to your other files.

There are directories on three levels of file storage:

The directories in a file structure are indexed by the Master File Directory (MFD) of that structure.

The files in your directory area are indexed by your User-File Directory (UFD). Your UFD is designated by your project-programmer number enclosed in brackets. The following is a valid directory name:

[30,112]

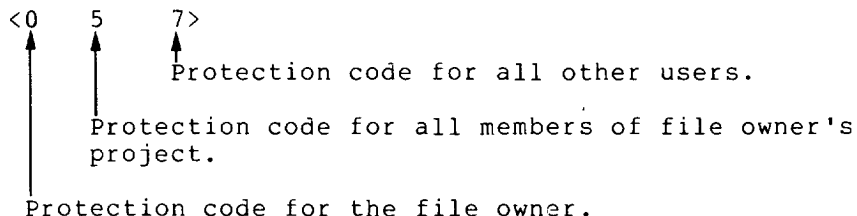
An SFD is a directory you can create to arrange the files in your UFD. A file in your UFD points to the SFD. This file has the same name as the SFD, with an extension of .SFD. SFDs are discussed in Section 1.12.

The directory name is an optional part of a file specification.

## INTRODUCTION

### 1.8.4 Protection Codes

Every file has a protection code. The code tells who can and cannot access the file. The protection code consists of three octal digits. Each digit specifies the amount of protection against a group of users. The first (leftmost) digit is the protection code for the owner of the file; the middle digit is the protection code for all users having the same project number as the file owner; the last (rightmost) digit is the protection code for all other users. For example:



Ordinarily, the owner of a file is the user whose programmer number matches the User File Directory (UFD) containing the file, regardless of the project number. That is, a user logged in under the [27,4072] project-programmer number is assumed to be the owner of files in the User File Directory [44,4072]. This feature can be set by the installation; therefore it may not be set at your own installation.

The access protection codes for the Owner Field (Field 1) are listed in Table 1-3; the access protection codes for Fields 2 and 3 are listed in Table 1-4.

Table 1-3  
Protection Codes for Field 1  
Owner Protection Codes

Code	Accessibility by Owner
7	You (the owner) have no access privileges. However, you can rename the file to change the protection code, using the RENAME command. If another user tries to access the file, the File Daemon* is called.
6	You can execute the file. If another user tries to write or rename the file, the File Daemon is called.
5	You can read and execute the file. If another user tries to write or rename the file, the File Daemon is called.
4	You can change the protection code of the file, rename, write, read, execute, update, and append to the file. If another user attempts to write, rename, or supersede the file, the File Daemon is called.

\*The File Daemon is discussed briefly in Section 1.9. For complete information, refer to the FILDAE Specification in the TOPS-10 Notebook Set.

## INTRODUCTION

Table 1-3 (Cont.)  
Protection Codes for Field 1  
Owner Protection Codes

Code	Accessibility by Owner
3	You can only rename the file.
2	You can write, read, execute, update, and append to the file.
1	You can rename, write, read, execute, update, and append to the file.
0	You can change the protection code of the file. You can also rename, write, execute, update, and append to the file.

Table 1-4  
Protection Codes for Fields 2 and 3  
Project-members and Other User Protections

Code	Access Privileges
7	Users other than the owner cannot access the file.
6	The user can execute the file. If the user attempts to delete, read, change, rename, append-to, or run the file with the RUN command, the File Daemon is called.
5	The user can read and execute the file. If the user attempts to write or rename the file, the File Daemon is called.
4	The user can read, execute, and append-to the file. If the user attempts to write, rename, or supersede the file, the File Daemon is called.
3	The user can read, execute, append-to, and update the file.
2	The user can write, read, execute, append-to, and update the file.
1	The user can rename, write, read, execute, append-to, and update the file.
0	The user can change the protection-code of the file. He can also rename, write, read, execute, change, append-to and update the file. This code allows all users full access privileges.

## INTRODUCTION

When you create a file, and you do not specify a protection code to be associated with the file, the system uses one of the following default protection codes:

- The standard TOPS-10 default protection code that is defined by your installation (usually <055> or <057>).
- The default protection code that is defined in your SWITCH.INI file on the LOGIN line, using the /DEFPROT switch.
- The default protection code you previously specified with the SET DEFAULT PROTECTION command.

### 1.8.5 File Daemon

The File Daemon is a system program that processes files that have a file protection code of 4, 5, 6, or 7 in the owner's protection field. Protection codes are discussed in Section 1.8.4. When such a protection is set, a protection violation causes the File Daemon to access a file in your UFD named ACCESS.USR. The file ACCESS.USR is a list of the protected files and the users who may access them. If you have no such file, or if the File Daemon is not running, the user who attempted to access the file receives an error message, and is not allowed to access the file. The File Daemon and the file ACCESS.USR are described in the FILDAE Specification in the TOPS-10 Notebook Set.

## 1.9 SYSTEM DEFAULTS

The operating system contains defaults for many arguments, switches and parts of file specifications. Defaults are the values or instructions that the system uses if you do not specify those values or instructions in the command string.

Some system defaults are set by the system administrator when he generates the monitor. Other defaults are set according to your project-programmer number when you log in. There are commands that allow you to set some defaults, which will be effective until you change those defaults or log off the system.

File specification defaults are determined by the system according to the program you are running, the search list, and the directory path of your job. Search lists and directory paths are discussed in Sections 1.11 and 1.12.

### 1.10 WILDCARD CONSTRUCTIONS

You can use wildcard constructions with many command strings. A wildcard is an asterisk used to specify a part of a file specification, or a question mark used to replace a character in a file specification field.

You use the asterisk (\*) as a wildcard to designate an entire part of a specification.

## INTRODUCTION

### Examples

All files with this file name and any extension:

```
file-name.*
```

All files with this extension and any file name:

```
*.ext
```

All files:

```
*.*
```

All files in directories with this project number and any programmer number:

```
*.*[project,*]
```

You can use the question mark as a wildcard to designate a character of a file specification. You type a question mark for each character that is to be matched. For example, PR?? matches four characters or less, of which the first two are PR.

### Examples

All files with this file name and any extension beginning with M:

```
file-name.M??
```

All files with this extension and any file name up to five characters, beginning with TES:

```
TES??.ext
```

All files with file names of two characters or less and a file name extension of three characters or less:

```
??.???
```

All files in directories with the project number 25 and a programmer number 500 through 577:

```
*.*[25,5??]
```

You can specify the asterisk and the question mark in the same command construction:

All files with file names of two characters or less:

```
??.*
```

The DIRECT and QUEUE programs recognize the number sign (#) to signify that the following character is to be translated literally. For example, if you typed \*.MAC as a file name in a command string, you would reference all the files with the .MAC extension. You can use the number sign followed by the octal representation of the file name or extension in SIXBIT code to type a file name or extension composed of special characters. For example, #12000000000 represents the file named \*. If letters or digits are part of the file name or the extension containing the special characters, you must also type the octal representation of the letters or digits following the number sign. Therefore, you must type the number sign at the beginning of the file name and all characters following it must be represented as the octal values for the characters.

## INTRODUCTION

You can specify a directory name with the project number, the programmer number, or both numbers missing from the specification. The following examples represent directory specifications.

[15,23]	The User-File Directory [15,23]
[,30]	The UFD that has your project number and the specified programmer number (30)
[36,]	The UFD that has the specified project number (36) and your programmer number
[,]	Your UFD
[-]	Your default directory, which can be different from your UFD. (See Section 1.11 and the SETSRC Program description in Chapter 2.)
[,,SUB1,SUB2]	The sub-file directory SUB2 under the sub-file directory SUB1 in your UFD

### 1.11 JOB SEARCH LIST

DSK: The job search list is a list of the file structures that the system uses to find a file. You specify the file structure in dev: of the file specification.

The generic structure DSK: represents your job search list. To examine or change your job search list, run the SETSRC program. SETSRC is described in Chapter 2 of this manual.

The job search list is divided into 2 parts, the active search list and the passive search list. The format of the search list is:

```
fs/switch,fs/switch,...FENCE,fs/switch...
```

Where: fs is the file structure name,

FENCE is a logical delimiter to separate the active search list from the passive structures.

/switch is the switch indicating how the file can be accessed (for example, read only).

The file structures listed to the left of the FENCE are the active search list. The system searches each of these file structures in the order of left to right.

The passive search list is to the right of the FENCE. The file structures in the passive search list are not searched by the monitor, but it is used to compute disk usage when you log off the system. The passive search list is comprised of the file structures that you removed from the active search list using the SETSRC program.

The default search list is set by the system administrator when you receive your project-programmer number. The default search list is only the active search list.



## INTRODUCTION

Your search list is also changed when you mount a new file structure. When you use the MOUNT command to access a file structure, the file structure is temporarily added to your search list. The file structure remains in your search list until you DISMOUNT the file structure, or until you log off the system.

Chapter 2 contains descriptions of the SETSRC program, and the MOUNT and DISMOUNT commands.

### 1.12 DIRECTORY PATHS

A directory path is an ordered list of directory names (starting with a user-file directory) that uniquely identifies a directory, regardless of file structure. Directory names are discussed in Section 1.8.3.

The directory path for your job can be your job's user-file directory (UFD), or sub-file directory (SFD) in your job's UFD, a UFD different from your job's UFD, or an SFD in a different UFD.

A sub-file directory is similar to a UFD, in that it acts as an index to a set of files. The SFD is pointed to by a file in the UFD, or by a higher-level SFD. You create the SFDs, and you can nest them in any structure, to the level that is predetermined by the system administrator. The maximum level to which SFDs can be nested is 6. SFDs are created by the CREDIR program, which is described in Chapter 2.

Sub-file directories allow you to arrange your files, and to access a set of files easily. Files that are written into a sub-file directory are no longer listed in the UFD. However, the SFD is designated by a file in the UFD with a name that you specify, and the extension .SFD.

Your directory path allows you to access a file in your UFD, other UFDs, or in SFDs, restricted by the file's protection code.

The following example shows the creation and use of an SFD and directory paths.

#### Example

Show a list of all the files in your UFD with a file extension .TST.

```
.DIRECT *.TST 
```

NUMB	TST	0	<055>	27-DEC-79	DSKC:	[27,5434]
109	TST	5	<055>	21-NOV-79		
FILL	TST	1	<055>	30-NOV-79		
PAY3	TST	1	<055>	12-DEC-79		

TOTAL OF 7 BLOCKS IN 4 FILES ON DSKC: [27,5434]

Run the CREDIR program.

```
.R CREDIR 
```

CREATE DIRECTORY: [27,5434,TEST]

CREATED DSKC:[27,5434,TEST].SFD/PROTECTION:775

## INTRODUCTION

Create a sub-file directory called TEST. Then exit from the CREDIR program.

```
CREATE DIRECTORY: ^C
```

Request a list of all your files named TEST.

```
.DIR TEST.*  
TEST   FOR   1   <055>   10-SEP-79       DSKC: [27,5434]  
TEST   SFD   1   <775>   27-DEC-79  
TOTAL OF 2 BLOCKS IN 2 FILES ON DSKC: [27,5434]
```

The directory shows an SFD named TEST.

Use the RENAME command to transfer all files with the extension .TST from your UFD to your SFD.

```
.RENAME [27,5434,TEST]=*.TST  
FILES RENAMED:  
DSKC:NUMB.TST  
DSKC:109.TST  
DSKC:FILL.TST  
DSKC:PAY3.TST
```

Show that your UFD no longer lists the files with the extension .TST.

```
.DIR *.TST  
%WLDNSF NO SUCH FILES AS DSKC:*.TST[27,5434]
```

Show that the files have been transferred to the SFD [27,5434,TEST].

```
.DIR [27,5434,TEST]  
NUMB   TST   0   <055>   27-DEC-79       DSKC: [27,5434,TEST]  
109    TST   5   <055>   20-NOV-79  
FILL   TST   1   <055>   30-NOV-79  
PAY3   TST   1   <055>   12-DEC-79  
TOTAL OF 7 BLOCKS IN 4 FILES ON DSKC: [27,5434,TEST]
```

CHAPTER 2  
SYSTEM COMMANDS AND PROGRAMS

Not all TOPS-10 commands run on all system configurations. Also your system administrator can add and delete commands from TOPS-10. Therefore, although this chapter contains detailed information on all of the TOPS-10 commands and some programs, not all of them may be available to you.

**2.1 FUNCTIONAL GROUPS OF COMMANDS**

In Sections 2.1 through 2.1.10, the TOPS-10 commands and programs are divided into functional groups. Each section contains:

- A discussion of a functional group.
- A list of the commands that make up the functional group.

Section 2.2 contains a detailed description of each command. The commands are arranged in alphabetical order.

Appendix A of this manual contains a table that lists the commands by functional description, including a short description of each command. This table is included for reference, so that you can find a command by its function. Please read the description of each command, which is in Section 2.2, before using the command.

This manual describes some system programs as well as the TOPS-10 commands.

**2.1.1 Job-Control Commands and Programs**

Job control commands control the state of your job. You can use them to create, detach, and terminate your job. Also, you can change the accounting profile and privilege status of your job. The job control commands are:

ATTACH	DETACH	DISABLE
ENABLE	KJOB	LOGIN
OPSER	REATTACH	

## SYSTEM COMMANDS AND PROGRAMS

### 2.1.2 Information Commands and Programs

The information commands allow you to gain information from the system. You can obtain information about your job, your program, the operating system, or your terminal characteristics. The information commands are:

CORE	CPUNCH	DAYTIME	DIRECTORY
DSK	HELP	INITIA	LOCATE
NETWORK	NODE	PJOB	PLOT
PRINT	QUEUE	QUOLST	RESOURCES
SCHEDULE	SET WATCH	SYSTAT	TIME
TPUNCH	USESTAT	VERSION	WHERE

### 2.1.3 Terminal-Control Commands and Programs

Terminal-control commands allow you to set the characteristics of your terminal, and to see the characteristics that are set for your terminal. The terminal-control commands are:

INITIA	SET TERMINAL
SET TTY	TERMINAL
TTY	

### 2.1.4 Terminal-Communication Commands

You can use the operating system to send a message to or to communicate with another terminal. The commands that allow you to do this are:

PLEASE
SEND

### 2.1.5 File-Handling Commands and Programs

You can use the operating system to create, change, and store files. You can create and change directories. You can output files to various devices. The file-handling commands and programs are:

BACKUP	CLOSE	COPY
CPUNCH	CREDIR	DELETE
DIRECTORY	DUMP	EOF
FILE	LIST	MAKE
PLOT	PRESERVE	PRINT
PROTECT	QUEUE	RENAME
SET DEFAULT PROTECTION	SETSRC	SUBMIT
TECO	TPUNCH	TYPE
ZERO		

## SYSTEM COMMANDS AND PROGRAMS

### 2.1.6 Device-Handling Commands

You can use the operating system to control peripheral devices. The device-handling commands are:

ASSIGN	BACKSPACE	CANCEL
CLOSE	CPUNCH	DEASSIGN
DISMOUNT	EOF	FINISH
LABEL	LIST	MOUNT
PRINT	QUEUE	REASSIGN
REWIND	SET BLOCKSIZE	SET CDR
SET DENSITY	SET RETRY	SET SPOOL
SKIP	TPUNCH	UNLOAD

### 2.1.7 Program-Preparation Commands and Programs

The program-preparation commands help you to write a program, change it, debug it, and obtain information about it. These commands help you to run programs more easily and effectively. The program-preparation commands and programs are:

CLOSE	COMPILE	CREF
D(eposit)	DCORE	DDT
DEBUG	E(xamine)	EOF
FILCOM	FUDGE	GLOB
LOAD	MAKE	MERGE
SET BREAK	TECO	

### 2.1.8 Program-Control Commands and Programs

Program-control commands help you to control your program while it is running and after it has been run. These commands are used to start and stop execution, save the core image, manipulate the core area, and to facilitate the execution of your program. The program-control commands are:

CCONTINUE	CONTINUE	CORE
CSTART	CTRL/C (HALT)	EXECUTE
GET	JCONTINUE	MERGE
NSAVE	NSSAVE	OSAVE
OSSAVE	R	REENTER
RUN	SAVE	SSAVE
SET CPU	SET DEFAULT BUFFERS	SET DSKFUL
SET DSKPRI	SET FLOATING POINT	SET HPQ
SET PHYSICAL	SETSRC	SET TIME
SET VIRTUAL	START	

### 2.1.9 Network Commands

Network commands help you to use a data network system. They allow you to use the resources on another system, and get information about the network configuration. The network commands are:

ASSIGN	LOCATE
NETWORK	NODE
SET HOST	WHERE

## SYSTEM COMMANDS AND PROGRAMS

### 2.1.10 MIC Commands

MIC (Monitor Interpreted Commands) allows you to create a new command by writing any desired sequence of monitor and MIC commands in a command file. Some MIC commands are briefly described here, and are more fully described in the file MICV2.DOC.

DO	Executes a MIC command file.
BACKTO GOTO	Specifies a label at which MIC processing is to resume within the command file.
ERROR/ NOERROR	Specifies an error condition character.
IF	Conditionally processes a monitor command.
LET	Changes the values of user parameters.
OPERATOR/ NOOPERATOR	Introduces a line requiring user attention.
SILENCE/ REVIVE	Suppresses/resumes output to the terminal.
WHENEVER/ ON	Changes the default action wherever a particular event occurs.

## 2.2 COMMAND DESCRIPTIONS

**ASSIGN Command****Function**

The ASSIGN command allocates an input/output device to your job until you log out or until you issue a REASSIGN, FINISH, or DEASSIGN command. (Device names are discussed in Section 1.8.1.) When an assignment is performed, the system prints a message indicating the name of the physical device assigned to your job.

You can ASSIGN any unrestricted device to your job.

**Formats**

ASSIGN device-name: logical-name:

Where: **device-name:** is any physical device name or previously assigned logical name.

(Refer to Section 1.8.1 for a description of the device-name formats.)

**logical-name** is a logical name that you request to be assigned to the physical device. Separate the physical device name from the logical name with a space. This argument is optional. Subsequent ASSIGN commands replace the old logical name with the new one. Logical names are disassociated from the devices when you issue the DEASSIGN command.

ASSIGN node-iddevice-name: logical-name:

Where: **node-id** is an identifier of the node from which the device is to be assigned. This identifier can be either a node name or a node number.

The node-id and the device name must be separated by an underscore (\_).

**device-name** can be any of the devices listed below:

Device	Generic Device Name
Card reader	CDR:
Line printer	LPT:
Task	TSK:
Terminal	TTY:

## ASSIGN Command (Cont.)

### Characteristics

The ASSIGN command:

Leaves your terminal at monitor level.

Does not destroy your core image.

### Associated Messages

If the assignment is successful, the system prints a message in the following format:

```
device-name: ASSIGNED
```

where device-name is the physical device-name specification. If you specify a restricted device, the monitor prints the following message:

```
?DEVICE NOT ASSIGNABLE
```

You can access restricted devices using the MOUNT command.

### Examples

1. Assign line-printer number 2.

```
.ASSIGN LPT2: (RET)
```

```
LPT262 ASSIGNED
```

.

2. Assign the logical name SYS to DSKB.

```
.ASSIGN DSKB: SYS: (RET)
```

```
DSKB: ASSIGNED
```

.

3. Assign the logical name TAPE to DTA5.

```
.ASSIGN DTA5: TAPE (RET)
```

```
DTA5: ASSIGNED
```

.

4. Assign a card-reader from node COMET to your job.

```
.ASSIGN COMET_CDR: (RET)
```

```
CDR701: ASSIGNED
```

.



## ATTACH Command

### Function

The ATTACH command detaches your current job (if you are logged in) and connects your terminal to a detached job.

To prevent unauthorized access to detached jobs, the monitor temporarily creates a new job when you specify the project-programmer number argument. This temporary job runs LOGIN, which asks for your password. If the temporary job exceeds the system's maximum job capacity, you may be unable to attach to the specified job. In this case, your first job remains detached. You will be unable to ATTACH to any job until there is room in the system.

### Format

ATTACH job [proj,prog]

Where: job is the job-number of the job to which your terminal is to be attached. This argument is required.

[proj,prog] is the project-programmer number of the desired job. You must enclose the PPN in square brackets ([]). If you are trying to attach from one job to another, and both have the same PPN, you can omit the project-programmer number argument.

### Characteristics

The ATTACH command:

Does not destroy the core image of either job.

Does not require that you be logged in.

Leaves your terminal at monitor level.

### Restrictions

Remote users cannot attach to jobs with a project number of 1.

Batch programs cannot use this command.

### Examples

1. Look at jobs 1 and 5 with SYSTAT.

```
.SYSTAT 1 5 (RET)
1    27,5434    DET    QUOLST    36+62    to    6 #
#MEANS NON-SYSTEM HI-SEG
5    27,5434    TTY31  SYSTAT    19+SPY    RN    25 $
$ MEANS EXECUTE ONLY
```

Output shows that job 1 is detached, and job 5 is attached to terminal number 31.

SYSTEM COMMANDS AND PROGRAMS

**ATTACH Command (Cont.)**

You attach to job 1 from job 5.

```
.ATTACH 1 (RET)
FROM JOB 5
```

The two jobs have the same project-programmer number; therefore, neither the project-programmer number nor the password is required.

Check jobs 1 and 5 again.

```
.SYSTAT 1 5 (RET)

1      27,5434   TTY31   SYSTAT   19+SPY   RN   25 $
$ MEANS EXECUTE ONLY
5      27,5434   DET     SYSTAT   24+SPY ^C   23 $
$ MEANS EXECUTE ONLY
```

Job 1 is now attached to TTY31, and job 5 is detached.

2. You log in to the system.

```
.LOGIN 27,235 (RET)

JOB 7 R5743A SYS #40/2 TTY116
PASSWORD: (RET)
10:34   17-JUN-1980   TUE
```

You are given job 7; terminal number 116 is now attached to job 7.

You attach to an existing job (35) and detach your current job (7). Because the project-programmer number associated with job 35 is different from your project-programmer number, you must specify the project-programmer number associated with job 35. LOGIN then requests that you enter the password. Enter the correct password. LOGIN attaches your terminal to job 35.

```
.ATTACH 35 [50,27] (RET)
PASSWORD: (RET)
```

You attempt to attach to job 7. The command fails because the project-programmer number belonging to job 7 is not the same as the project-programmer number for job 35. Your terminal is still attached to job 35.

```
.ATTACH 7 (RET)
?CAN'T ATTACH TO JOB
```

```
.K/F (RET)
JOB 35, USER MAROTTA [50,27]
LOGGED OFF TTY116 AT 12:03:36
DISK READS: 78, WRITES: 344, BLOCKS SAVED: 7998
```

## ATTACH Command (Cont.)

You log out of job 35. Your terminal is not attached to any job.

```
.ATTACH 7   
?CAN'T ATTACH TO JOB
```

Because your terminal is currently not attached to a job, your command fails. There is no project-programmer number to compare with the project-programmer number associated with job 7.

```
.ATTACH 7 [27,235]   
PASSWORD: 
```

The monitor accepts the command and LOGIN requests you type the password. The monitor does not print the message

```
FROM JOB n
```

because your terminal was not attached to any job. Then LOGIN attaches your terminal to job 7.

## BACKSPACE Command

### Function

The BACKSPACE command spaces a magnetic tape backward a specified number of files or physical records. This command runs the COMPIL program, which interprets the command before running PIP.

### Formats

BACKSPACE MTAn: x FILES

This command skips backward x files.

BACKSPACE MTAn: x RECORDS

This command skips backward x records.

### Characteristics

The BACKSPACE command:

Leaves your terminal at monitor level.

Runs the PIP program.

Destroys your core image.

### Examples

1. Backspace 7 records on the tape on MTA2.

```
.BACKSPACE MTA2: 7 RECORDS (RET)
```

2. Backspace 11 files on the tape on MTA3.

```
.BACKSPACE MTA3: 11 FILES (RET)
```

## BACKUP Program

### Function

The BACKUP program saves disk files on magnetic tape and places all or some of these files back onto the disk. You can save your disk area on magnetic tape and restore all of your files or a subset of your files back to your area. You can also use this program to move your files from one system to another.

### Format

```
R BACKUP
/command
```

Where: **command** is one of the BACKUP commands, which are in the form of verbs. BACKUP prompts with a slash (/) after which you respond with one of three types of verbs:

- Action verb
- Status-setting verb
- Tape-positioning verb

### Action Verbs

Action verbs perform I/O on the tape specified in the last TAPE command, which is one of the status-setting verbs. You must specify the tape before using an action verb.

SAVE file-spec	Saves the specified disk files on tape.
RESTORE file-spec	Restores the specified tape files to disk.
CHECK file-spec	Verifies that the tape and the disk files agree.
[N]PRINT file-spec	Prints a directory of the entire tape specified by file-spec. N is an optional prefix indicating narrow. PRINT implies a NODIRECTORIES command; that is, no user directory names are printed.

The file-spec arguments are optional. You can specify input and output files in the form: output=input, or you can specify input alone. The default input file-spec is ALL:\*. \*[PPN,\*,\*,...].

### Action Verb Switches

You can use the following switches in the file specification list for the action verbs. These switches can be either permanent or temporary. (Refer to Section 1.7.4.)

## SYSTEM COMMANDS AND PROGRAMS

### BACKUP Program (Cont.)

/ABEFORE date	On input file, includes only files accessed before the specified date.
/ASINCE date	On input file, includes only files accessed since the specified date.
/BEFORE date-time	On input file, includes only files created before the specified date-time.
/ERNONE	On input file, gives error if no files match.
/ERPROTECTION	On input file, gives error if there is a protection failure.
/ERSUPERSEDE	On output file, does not restore from tape if on disk.
/ESTIMATE n	On output file, estimates output size (n).
/LENGTH l:h	On input file, includes only if file length is between l (low address) and h (high address).
/MBEFORE nD:hh:mm:ss	On input file, includes only files modified before date-time.
/MSINCE nD:hh:mm:ss	On input file, includes only files modified since date-time.
/NOPHYSICAL	On input or output file, uses the logical names.
/OKNONE	On input file, does not give an error if no files match.
/OKPROTECTION	On input file, does not give an error if there is a protection failure.
/OKSUPERSEDE	On output file, always restores, even if the file is already on disk.
/PHYSICAL	On input or output file, ignores any logical names.
/PROTECTION nnn	On output file, sets the protection code.
/SINCE date-time	On input file, includes only if created since the specified date-time.
/STRS	On input file, looks at all structures for file.
/VERSION v	On output file, sets version number of the file.

**BACKUP Program (Cont.)**

**Runtime Commands**

The following runtime commands can be given during the execution of the action verbs. BACKUP prompts with an exclamation point (!) when ready to accept a runtime command.

DIRECTORIES	Prints every directory processed.
EXIT	Exits from BACKUP when done.
FILES	Prints every file and directory processed.
GO	Continues after a STOP.
HELP	Lists these commands and explanations.
KILL	Aborts execution of the current action verb.
NODIRECTORIES	Does not print every directory processed.
NOFILES	Does not print every file in every directory processed.
PAUSE	Does not exit from BACKUP when done.
RESET	Resets all status settings to their original defaults when done.
SILENCE	Stops printing every directory or file.
STOP	Stops temporarily; can be continued by GO.
WHAT	Displays current file name and status.

**Status-Setting Verbs**

Status-setting verbs specify a parameter(s) that affects future action verbs.

ABEFORE date	Includes only files accessed before the specified date.
APPEND	Appends to the existing list file.
ASINCE date	Includes only the files accessed since the specified date.
BEFORE date-time	Includes only the files created and last modified before the specified date-time.
CPOINT	Uses checkpoints.
DATE75	Always accepts files with possible DATE75 problems.
DELETE	Deletes disk files after saving them.
DENSITY nnnn	Sets the tape density. The density can be 200, 556, 800, 1600, or 6250.

## SYSTEM COMMANDS AND PROGRAMS

### BACKUP Program (Cont.)

DIRECTORIES	Prints each user's directory while running.
ENCRYPT	Encrypts (codes) the files. This makes the files impossible to read without the keyword.
EXEMPT	Exempts project, programmer numbers of the form [A,*] and [*,B], where A and B are less than 7, from date, time and length restrictions.
FILES	Prints each file name while running.
INITIAL file	Starts processing at the specified file name.
INTERCHANGE	Runs in interchange mode. Interchange is used to copy files from a TOPS-20 processor or from one TOPS-10 installation to another.
LENGTH low:high	Includes only files whose length is between l (low address) and h (high address).
LIST spec	While running, makes a listing file on spec. LIST produces the effect of a NODIRECTORIES command; that is, no user directories will be printed.
MBEFORE date-time	Includes only files that were moved or modified before the specified time.
MESSAGE [NO]PREFIX	Suppresses the error message prefix.
MESSAGE [NO]FIRST	Suppresses the first line of the error message text.
MSINCE date-time	Includes only files that have been moved or modified since the specified time.
MULTIREEL	Allows multiple reels during a save.
NOAPPEND	Does not append to the existing list file.
NOCPOINT	Does not use checkpoints.
NODATE75	Does not accept files with possible DATE75 problems.
NODELETE	Does not delete disk files after saving them.
NODIRECTORIES	Does not print each user's directory while running.
NOENCRYPT	Does not encrypt (code) the files.
NOEXEMPT	Does not exempt PPN's of [A,*] and [*,B] from restrictions.
NOFILES	Does not print each file while running.



## SYSTEM COMMANDS AND PROGRAMS

### BACKUP Program (Cont.)

NOINTERCHANGE	Does not run in interchange mode.
NOLIST file	Does not make a listing file.
NOMULTIREEL	Does not allow multiple reels during the save.
NOREPEAT	Does not repeat a split file on the continuation tape.
NOUSETI	Does not run in USETI mode.
NOWRITE	Suppresses disk writing during a restore.
OPTION name	Uses line BACKUP:name from a SWITCH.INI file.
PARITY (EVEN,ODD)	Sets the tape parity.
REPEAT	Repeats a split file on the continuation tape.
RESUME n	Resumes at block n of the initial file.
SILENCE	Does not print file names or directories while running.
SINCE date-time	Includes only those files created or modified since the specified date-time.
SORT DIRECTORIES x	Sorts directories within each file structure in order of x when saving. (x is ALPHABETICAL, LOCATION, or NONE.)
SORT FILES x	Sorts files within each directory in the order of x. (x is ALPHABETICAL, LOCATION, or NONE.)
SSNAME name	Specifies the save-set name.
SUPERSEDE ALWAYS	Restores all files from tape, superseding files on disk that have the same file name.
SUPERSEDE NEVER	Does not restore files that have the same name as an existing file.
SUPERSEDE OLDER	Restores only the files that are newer than the files on disk of the same name.
TAPE x	Uses tape unit x. If a magnetic tape drive has the logical name BACKUP, then the TAPE verb need not be specified.
TPNUM x	Sets the number of the tape to x (decimal).
UPROTECTION nnn	Sets the protection code for the created directories.
USETI	Runs in USETI mode.
WRITE	Writes onto disk during a RESTORE. This is the default.

## BACKUP Program (Cont.)

### Tape-Positioning Verbs

EOT volid-list	Skips to the end-of-tape mark for each tape in the list. The volid-list is a list of the volume-identifications of each tape.
REWIND volid-list	Rewinds to the beginning of the tape for each tape in the list.
SKIP n volid-list	Skips the specified number of save sets forward for each tape in the list.
SKIP 0 volid-list	Backs up to the start of the current save set for each tape in the list.
SKIP -n volid-list	Skips the specified number of save sets backward for each tape in the list.
UNLOAD volid-list	Unloads each tape from its drive for each tape in the list.

### Characteristics

The R BACKUP command:

Places your terminal at user level.

Destroys your core image.

### Examples

- To save the entire contents of DSKB: on magnetic tape and at the same time produce a directory listing, run BACKUP while logged in under [1,2].

```
.R BACKUP (RET)
/TAPE MTA0 (RET)
/LIST DSK:BACKUP.LOG (RET)
/SAVE DSKB: (RET)
!1,2 DSKB
 1,3
 1,4
.
.
.
.
```

BACKUP prints each UFD as it begins to save files from that area. If the tape becomes full before the save is complete, BACKUP stops, types out the full file specification and block number of the current file being saved, unloads the magnetic tape and prints the following message:

```
$BKPEOT REACHED EOT -- MOUNT NEW TAPE THEN TYPE "GO"
```

When you type GO, BACKUP continues the save on the new tape. A file can be split across tapes.

SYSTEM COMMANDS AND PROGRAMS

**BACKUP Program (Cont.)**

2. Copy the file FILE.MAC from tape to disk, without changing its device and directory specification.

```
.R BACKUP (RET)
/TAPE MTB0 (RET)
/REWIND (RET)
/RESTORE FILEB.MAC=FILEA.MAC (RET)
!10,123 DSKB
DONE
/UNLOAD (RET)
/^C
```

## CCONTINUE Command

### Function

The CCONTINUE command starts program execution, leaving your terminal at monitor level.

### Format

CCONTINUE

To use:

1. Begin running a program.
2. Exit from user mode by typing two CTRL/Cs.
3. Type CCONTINUE to allow the previously begun program to continue running from the point at which you interrupted it. Your terminal is left at monitor level.
4. You can now use commands that do not require core, or you can detach from your job and create a new job to run other programs.

### Characteristics:

Leaves your terminal at monitor level.

Requires core.

### Associated Commands

CONTINUE - Continues the operation of your program, bringing your terminal back to user level.

START - Starts the program from the beginning or from the specified address, leaving your terminal at user level.

CSTART - Starts the program from the specified address or from the beginning, but leaves your terminal at monitor level.

### NOTE

If your program requires terminal I/O, the CCONTINUE command allows the program to run only to that point. The program then waits for terminal I/O, before it continues executing. Use the CONTINUE command to re-enter user level and accept or input the required I/O.

## CCONTINUE Command (Cont.)

## Example

This is a program that finds all the numbers up to 10,000 and writes them out to disk.

```
.TYPE NUMBER.FOR(RET)

      N = 0
100   N = N + 1
      IF (N .EQ. 10000) GO TO 300
      WRITE (22, 201) N
      GO TO 100
201   FORMAT (1X, I14, ' IS BETWEEN 1 AND 10000')
300   STOP 'DONE'
      END
```

Execute the program.

```
.EXECUTE NUMBER.FOR(RET)
FORTRAN: NUMBER
NUMBER
LINK:   LOADING
[LNKXCT NUMBER EXECUTION]
^C
^C
```

Type two CTRL/C's to halt the program.

Continue the program, leaving your terminal at monitor level.

```
.CCONTINUE(RET)
```

Type CTRL/T for job status. CTRL/T does not echo on your terminal.

```
DAY: 8.85 RUN: 2.05 RD:0 WR:0 NUMBER 4+15P RN* PC:000175
```

The status message RN\* indicates the program NUMBER is running.

Detach from the job to do work on another job.

```
.DETACH(RET)
```

```
FROM JOB 19
```

```
.
```

```
.
```

```
.
```

Later, attach to your original job.

```
.ATTACH 19 [27,5434](RET)
```

```
PASSWORD:(RET)
```

Type CONTINUE to re-enter user level.

```
.CONTINUE(RET)
```

**CCONTINUE Command (Cont.)**

System message (see NOTE).

?PLEASE TYPE ^C FIRST

Type CTRL/C and CONTINUE, to enter user level.

.^C

.CONTINUE

Program message indicates execution is finished. The file containing data from this program is named FOR22.DAT, and is stored in your default disk area.

DONE

END OF EXECUTION

CPU TIME:4.60 ELAPSED TIME:12:12.83

EXIT

.

NOTE

It is necessary to type CTRL/C to re-enter user level, because the program is running. The program must be interrupted so that your terminal can access it.

## CLOSE Command

### Function

The CLOSE command terminates any input or output currently in progress on the specified device.

Although most programs close files when they finish executing a command string, the CLOSE command is provided for a program that does not terminate or a program that is being debugged. This command causes all files to be written into your UFD. If a CLOSE is not done, the next RESET by a command (R, RUN, GET) or program deletes any partially written disk files.

### Format

CLOSE dev:

Where: **dev:** is 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 your terminal at monitor level.

Requires core.

### Restrictions

You cannot continue the program after using CLOSE, but you can restart at the beginning of your program or you can access DDT.

### Examples

1. Terminate input from the paper-tape reader number 2.

```
.CLOSE PTR2:RET
```

.

2. Terminate I/O from the device with the logical name DEVA.

```
.CLOSE DEVA:RET
```

.

3. Terminate I/O from all devices except your terminal.

```
.CLOSE:RET
```

.

**COMPILE Command****Function**

The COMPILE command produces relocatable binary files (.REL files) and/or compilation listings for specified source program files. The system determines which language compiler to use by the source file extension or by switches you specify in the command string. If you do not supply any switches in the command string, but you do use a standard extension, the system uses the following compilers:

Source File Extension	Compiler Used
.ALG	ALGOL compiler
.BLI, .BlO	BLIS10 compiler
.CBL	COBOL compiler
.F4 or .FOR	FORTRAN compiler
.FAI	FAIL compiler
.FCL	FOCAL compiler
.LSP	LISP compiler
.MAC	MACRO assembler
.Pl1	MACY11 assembler
.SAI	SAIL compiler
.SNO	SNOBOL compiler

**NOTE**

The compiler cannot be changed with a switch, but it can be specified with a switch when the source file has an unrecognizable or no extension. If the source file has a non-standard extension, and you do not specify the compiler in a switch, FORTRAN is used as the default compiler. All standard file extensions are listed in Appendix D.

Usually, the system translates the source file if there is no corresponding binary (.REL) file, or if the date and time of the source file is later than the date and time of the binary file. If the binary file is newer than the source file, the system does not generate a new .REL file.

This command runs the COMPIL program, which interprets the command before running the appropriate language compiler.

FAIL, FOCAL, LISP, MACY11, SAIL, and SNOBOL are recognized as compilers only if the appropriate assembly switches are set at your installation.

Each time you issue the COMPILE, LOAD, EXECUTE, or DEBUG commands, the system remembers the command, with its arguments, in a temporary file on disk or in TPCOR if they are small enough. When you issue one of these commands without an argument, it causes the system to use the argument saved in the temporary file.



**COMPILE Command (Cont.)**

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

**Format**

COMPILE file-spec/switch,file-spec/switch,...

Where: **file-spec** is a single file specification or a string of file specifications, separated by commas. The standard file specification consists of: a device name, a file name (with or without an extension), and a directory name. If you omit the file specification argument, the system uses the argument specified in a previous COMPILE-class command, if possible. (Refer to Section 1.8.)

You can use the following switches as temporary or permanent switches. (Refer to Section 1.7.4.) Switches relevant to COMPILE follow; all switches allowed with COMPILE can be used with LOAD, EXECUTE and DEBUG.

Switch	Function
/ALGOL	Compiles with ALGOL. Assumed for files with the .ALG extension.
/BINARY	Generates a binary file for each file compiled. The file name of the binary file is .REL. This is the default action.
/BLISS	Compiles the file with BLISS10. Assumed for files with the extension of .B10 and .BLI.
/C68	Compiles with COBOL 68.
/C74	Compiles with COBOL 74.
/COBOL	Compiles the file with COBOL. Assumed for files with the extension of .CBL.
/COMPILE	Forces a compilation of this file even if a binary file exists with a newer date and time than the source file. Use this switch to obtain an extra compilation (for example, to obtain a listing of the compilation) because, normally, compilation is not performed if the binary file is newer than the source file.
/CREF	Produces a listing file on the disk for each file compiled, for later processing by the CREF program. The file extension of the listing file is .CRF. The file can then be listed with the CREF command. With COBOL files, the CREF file is appended to the output file; additional commands are needed to obtain the cross-referenced file.
/DDT	Loads the program debugger DDT with the file.

## COMPILE Command (Cont.)

Switch	Function
/DEBUG:(arg, arg,...)	Passes the arguments to FORTRAN. Refer to the FORTRAN-10 Programmer's Reference Manual.
/DLIST	Creates a file with the extension .LST and stores it in your disk area. You can have this file printed on the line printer by using the PRINT command.
/F40 /F10	Obsolete
/FORTRAN	Compiles the file with a FORTRAN compiler. Assumed for files with either the extension of .F4 or .FOR and for all files with unrecognizable compiler extensions, if FORTRAN is the standard compiler for your system. This switch is necessary if the file has a unrecognizable compiler extension and FORTRAN is not the standard compiler or is not the current default.
/FORDDT	Loads the FORDDT debugger with the program.
/FOROTS /FORSE	Obsolete
/FUDGE:file	Creates a disk file that contains the names of the .REL files produced by the command string. When you issue the FUDGE command, PIP reads this file to generate a library .REL file. (See the MAKLIB User's Guide in the TOPS-10 Notebook Set for information about library files.)  This switch requires a file specification argument. The file name is necessary. If you do not specify a device, the default is DSK:. If you omit the file extension, the default is .REL. If you omit the directory, the default is your own directory.  This switch is permanent. It pertains to all .REL files generated by this command string.
/K?10	Designates the machine on which the program will execute after it has been loaded. The default is the processor that is executing the command. The ? can be A, I, or L. To designate a KS processor, use /KL10.
/LIBRARY	Loads the program in library search mode.
/LINK	Obsolete
/LIST	Generates a listing file for each file compiled. The file name extension is .LST. These files can be printed later with the PRINT command. If the line printer is spooled for your job, the listing files are automatically printed.

## SYSTEM COMMANDS AND PROGRAMS

### COMPILE Command (Cont.)

Switch	Function
/LMAP:file	Produces a loader map while loading. The map contains local symbols.
/MACRO	Assembles the file with MACRO. Assumed for files with the extension of .MAC.
/MACY11	Assembles the file with MACY11. Assumed for files with the extension .Pl1. MACY11 is recognized as a compiler only if the appropriate assembly switch is set. This switch is not supported.
/MANTIS	Compiles the program with MANTIS debugging information. This switch affects FORTRAN-40 files only. MANTIS is recognized as the debugging program only if the appropriate assembly switch is set. This switch is not supported.
/MAP:file	Produces a loader map while loading. The default file name is MAP.MAP.
/NEW	Runs the appropriate language compiler from the experimental library area [1,5] (device NEW:). If the compiler does not exist on device NEW:, COMPIL tries to obtain it from device SYS:.
/NOBINARY	Does not generate binary files. This switch, when combined with the /CREF or /LIST switch, is useful when compiling programs solely to generate listings.
/NOCOMPILE	Does not force a compilation of a source file whose date is earlier than the date on the binary file. Complement to the /COMPILE switch. /NOCOMPILE is the default action.
/NODEBUG	Does not pass previously specified arguments to FORTRAN.
/NOLIST	Does not generate listing files. This is the default action. Complement to /LIST.
/NOMANTIS	Compiles the program without the MANTIS debugging information. This is the default. This switch affects FORTRAN-40 files only. Complement to /MANTIS. This switch is not supported.
/NOOPTIMIZE	Does not optimize the object source code. This is the default. Complement to /OPTIMIZE.
/NOSEARCH	Does not load the program in library search mode. Complement to /SEARCH.
/OPTIMIZE	Optimizes the object code of a FORTRAN program.

## SYSTEM COMMANDS AND PROGRAMS

### COMPILE Command (Cont.)

Switch	Function
/OLD	Runs the appropriate language compiler from the system library [1,3] of old programs (device OLD:). If the compiler does not exist on device OLD:, COMPIL tries to obtain it from device SYS:.
/REL	Loads an existing .REL file rather than creating a new one.
/SAVE	Saves the core image in an executable file after it is loaded.
/SEARCH	When the file is loaded, uses library search mode.
/SELF	Runs the appropriate language compiler from device DSU: instead of from the system library (device SYS:). This switch is useful if you keep a private copy of a compiler in your own disk area.
/SNOBOL	Compiles the file with SNOBOL. Assumed for files with the extension .SNO. SNOBOL is recognized as a compiler only if the appropriate assembly switch is set. This switch is not supported.
/SSAVE	Saves the core image in a sharable executable file after the program is loaded.
/SYS	Loads the program with the compiler from SYS:. This is the default.

#### Characteristics

The COMPILE command:

Leaves your terminal at monitor level.

Runs the appropriate language compiler, destroying your original core image.

#### Restrictions

You cannot use wildcard constructions with COMPILE.

A language compiler appearing more than once within a single command string cannot access more than one disk area. For example, the following is invalid:

```
.COMPILE TESPRG.F4/NEW,SUBRTN.F4/SYS
```

However, the following is valid:

```
.COMPILE TESPRG.F4/NEW  
.COMPILE SUBRTN.F4/SYS
```

**COMPILE Command (Cont.)****Examples**

1. Compile PROG (with the null extension) with FORTRAN, TEST.MAC with MACRO, and MANAGE (with the null extension) with MACRO. A listing file is generated for MANAGE. The files generated are PROG.REL, TEST.REL, MANAGE.REL, and MANAGE.LST.

```
.COMPILE PROG,TEST.MAC,MANAGE/MACRO/LIST (RET)
```

```
FORTRAN:  PROG
MAIN
MACRO:    TEST
MANAGE
EXIT
```

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

```
.COMPILE/LIST SIGN.MAC,TABLES/NOLIST,MULTI.ALG (RET)
```

```
MACRO:    SIGN
FORTRAN:  TABLES
MAIN
ALGOL:    MULTI
EXIT
```

3. Compile the files DIVIDE, SUBTRC, and ADD with the FORTRAN compiler, even if current .REL files exist. Generate files to be processed by CREF. The files generated are DIVIDE.CRF, DIVIDE.REL, SUBTRC.CRF, SUBTRC.REL, ADD.CRF and ADD.REL.

```
.COMPILE/CREF/COMPILE DIVIDE,SUBTRC,ADD (RET)
```

```
FORTRAN:  DIVIDE
MAIN.
FORTRAN:  SUBTRC
MAIN.
FORTRAN:  ADD
MAIN.
```

SYSTEM COMMANDS AND PROGRAMS

**COMPILE Command (Cont.)**

4. Compile the files contained in the command file LIBALL and write the names of all the .REL files generated in a temporary disk file. The name of the library file is MONITR.REL. The library file is created with the .REL files in the same order as they were specified in the command string.

```
.COMPILE/FUDGE:MONITR.REL@LIBALL (RET)
```

```
FORTRAN: DIVIDE  
MAIN.  
FORTRAN: SUBTRC  
MAIN.  
FORTRAN: ADD  
MAIN.
```

.

5. Compile and execute the program NUMBER.FOR.

```
.COMPILE NUMB2.FOR (RET)
```

```
FORTRAN: NUMB2  
MAIN.
```

```
.EXECUTE (RET)  
LINK: LOADING  
[LNKXCT NUMB2 EXECUTION]  
DONE  
END OF EXECUTION  
CPU TIME: 0.12 ELAPSED TIME: 0.53  
EXIT
```

.

**CONTINUE Command****Function**

The CONTINUE command continues your program from the point at which you interrupted it. You interrupt program execution with CTRL/C. After you use CONTINUE, your terminal returns to user level.

**Format**

```
CONTINUE
```

**Characteristics**

The CONTINUE command:

Places your terminal at user level.

Requires core.

**Example**

This is a program that finds all the numbers up to 10,000.

```
.TYPE NUMBER.FOR (RET)
      N = 0
100   N = N + 1
      IF (N .EQ. 10000) GO TO 300
      WRITE (22, 201) N
      GO TO 100
201   FORMAT (1X, I14, ' IS BETWEEN 1 AND 10000')
300   STOP 'DONE'
      END
```

Execute the program.

```
.EXECUTE NUMBER.FOR (RET)
FORTRAN:NUMBER
NUMBER
LINK:LOADING
[LNKXCT NUMBER EXECUTION]
^C
^C
```

Type two CTRL/Cs to halt the program.

Detach from the job to do work on another job.

```
.DETACH (RET)
FROM JOB 19
```

.  
.  
.

SYSTEM COMMANDS AND PROGRAMS

CONTINUE Command (Cont.)

Later, attach to your original job.

.ATTACH 19 [27,5434]

PASSWORD:

Type CONTINUE to enter user level.

.CONTINUE

?PLEASE TYPE ^C FIRST

System message (see NOTE).

Type CTRL/C and CONTINUE to enter user level.

.^C

.CONTINUE

DONE

END OF EXECUTION

CPU TIME:4.60 ELAPSED TIME:12:12.83

EXIT

.

Program message indicates execution is finished. The file containing data from this program is named FOR22.DAT, and is stored in your default disk area.

NOTE

It is necessary to type CTRL/C to re-enter user level, because the program is running. The program must be interrupted so that your terminal can access it.



## COPY Command

### Function

The COPY command copies a file from one device to another, or within a device, file structure, or directory. The command string can contain one output specification and any number of input specifications.

### Format

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

Where: The file to the left of the equal sign (=) is the destination, or output file, and the file(s) to the right of the equal sign is the source, or input file(s).

**dev:** is a physical or logical device name. If you omit a device name, the system assumes DSK:.

**file.ext** is the name of the file(s) to be used in input or output. If you omit the output file name, the system defaults to the input file name. If you transfer many input files to one output file, the system combines the files. You can use wildcard constructions with the COPY command.

**[directory]** is the disk area in which the files are to be read or written. If you type this area before the file name, the system uses this area as the default for all succeeding files. If you omit this argument, your default directory is accessed. You can transfer files to or from another directory only if you have access to that directory.

**<nnn>** is the protection code to be given to the output file. If you omit this argument, the system assigns the system standard protection code, even if the input file already has a non-standard protection code associated with it. Protection codes are described in Section 1.8.4.

Use the equal sign (=) to separate the destination (output) side from the source (input) side.

### Characteristics

The COPY command:

Leaves your terminal at monitor level.

Destroys your core image.

Runs the PIP program.

## COPY Command (Cont.)

### Example

Find file NOTICE.TXT in SYS:

```
.DIRECT SYS:NOTICE.TXT 
```

```
NOTICE TXT 2 <155> 6-SEP-79 DSKC:[1,4]
```

Search your own directory.

```
.DIRECT DSKB:NOTICE 
```

```
?NOTICE (0) FILE WAS NOT FOUND
```

Copy NOTICE from SYS: into your directory on DSKB:

```
.COPY DSKB:NOTICE.TXT=SYS:NOTICE.TXT 
```

Check your directory to find NOTICE on DSKB:

```
.DIR NOTICE 
```

```
NOTICE TXT 2 <055> 6-SEP-79 DSKB:[27,5434]
```

## CORE Command

### Function

The CORE command prints or changes the amount of core assigned to your job. Because programs usually allocate core, you generally do not need this command.

If your job is locked in core and you issue this command with a nonzero argument, the monitor prints an error message.

### Format

CORE nx

Where: n is a decimal number; this argument is optional.

If you omit n, the monitor prints the amount of core currently being used by your job, and does not change the core assignment.

If n is 0, the low and high segments are removed from the virtual addressing space of your job.

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

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

x is either K or P. K represents units of 1024 words. P represents 512-word pages. For example, 3P represents three pages or 1536 words. If you do not specify x, K is assumed.

### Characteristics

The CORE command:

Leaves your terminal at monitor level.

Does not operate when your job is running.

### Example

Use CORE to look at the amount of memory assigned:

```
.CORE RET
```

```
VIRTUAL MEMORY ASSIGNED 4+10P (CURRENT LIMIT:512P, MAXIMUM  
LIMIT:512P)  
PHYSICAL MEMORY ASSIGNED 4+10P (GUIDELINE:512P, MAXIMUM  
LIMIT:510P)  
SWAP SPACE LEFT:7849P
```

## SYSTEM COMMANDS AND PROGRAMS

### CORE Command (Cont.)

Use CORE to change the memory assignment:

```
.CORE 12K(RET)
```

Use CORE to look at the memory assignment:

```
.CORE(RET)
```

```
VIRTUAL MEMORY ASSIGNED 15+10P (CURRENT LIMIT:512P MAX  
LIMIT:512P)  
PHYSICAL MEMORY ASSIGNED 15+10P (GUIDELINE:512P MAXIMUM  
LIMIT:510P)  
SWAP SPACE LEFT:7805P
```

#### NOTE

This example specifies that the amount of core be increased to 12K. However, the output from the CORE command showed that the total area assigned equalled 25 pages. The one page discrepancy results from the per-process space. This is a non-addressable space that is used by the monitor to track your job. The per-process space can be any amount between one and sixteen pages.

## 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=file-spec/switches,file-spec/switches...
```

### Format

```
CPUNCH jobname=file-spec/switches,file-spec/switches,...
```

Where: **jobname** is the name of the job you are entering into the queue. The default jobname is the name of the first file in the request. This might not be the name of the first file appearing in the command string, if that file does not yet exist. If you omit the jobname, you can omit the equal sign (=).

**file-spec** is the file or files you want punched. File specifications are described in Section 1.8. You can use the wildcard constructions in the file-spec.

**/switches** is the switch or switches applying to that file. The switches are listed in the description of the QUEUE command.

You can obtain a list of the entries in the card punch output queue for a specific project-programmer number by following the command with the desired project-programmer number enclosed in square brackets (for example, CPUNCH [40,200]). If you omit all arguments to the command, all entries in the card punch queue are printed.

### Characteristics

The CPUNCH command:

- Leaves your terminal at monitor level.

- Destroys your core image.

- Does not require LOGIN when only a list of queue entries is desired.

### Example

Punch the file SYSTAT.MAC in ASCII format.

```
.CPUNCH SYSTAT.MAC/PUNCH:ASCII RET  
[LPT:SYSTAT=/SEQ:65/LIMIT:200/PRI0:10/PUNCH:ASCII]
```

## CPUNCH Command (Cont.)

Show a list of the entries in the card punch queue:

```
.CPUNCH   
DEV      JOB      SEQ      Prio      LIMIT     USER  
----      ---      ---      ----      -  
CDP      SYSTAT 65      10        200       MAROTTA  
/PUNCH:ASCII
```

Kill the card punch job:

```
.CPUNCH SYSTAT=/KILL   
[ONE JOB KILLED]
```

## CREATE Command

This command and the program LINED are obsolete and unsupported.

## CREDIR Program

### Function

The CREDIR program creates directories and sub-file directories. CREDIR creates sub-file directories (SFDs) on all specified structures, and it creates any higher-level directories that might be necessary.

You can use CREDIR to create ersatz device directories. CREDIR allows you to create directories on all structures associated with a specified ersatz device. As it creates each directory, CREDIR tells you the name of the unit and the protection code associated with the directory.

### Format

R CREDIR

CREDIR prints CREATE DIRECTORY:, after which you respond with the device name and the directory specification. You can specify an ersatz device name.

You can use the standard SCAN switches with the CREDIR program.

Device DSK: is assumed if you do not specify a device. If you do not specify a protection code for the directory, the system gives the directory a default protection code of <775>.

### Characteristics

The CREDIR program:

Destroys your core image.

Requires LOGIN.

### Examples

- The following example shows the creation and use of an SFD and directory paths.

Show a list of all the files in your UFD with a file extension .TST.

```
.DIRECT *.TST RET
```

```
NUMB  TST  0  <055>  27-DEC-80      DSKC:  [27,5434]
109   TST  5  <055>  21-NOV-80
FILL  TST  1  <055>  30-NOV-80
PAY3  TST  1  <055>  12-DEC-80
TOTAL OF 7 BLOCKS IN 4 FILES ON DSKC: [27,5434]
```

Run the CREDIR program.

```
.R CREDIR RET
```

```
CREATE DIRECTORY: [27,5434,TEST] RET
CREATED DSKC: [27,5434,TEST].SFD/PROTECTION:775
```



**CREDIR Program (Cont.)**

Create a sub-file directory called TEST. Use CTRL/C to exit from CREDIR.

CREATE DIRECTORY: ^C

Request a list of all your files named TEST.

```
.DIR TEST.* (RET)

TEST  FOR  1  <055>  10-SEP-79      DSKC: [27,5434]
TEST  SFD  1  <775>  27-DEC-80
      TOTAL OF 2 BLOCKS IN 2 FILES ON DSKC: [27,5434]
```

The directory shows an SFD named TEST.

Use RENAME to transfer all files with the extension .TST from your UFD to your SFD.

```
.RENAME [27,5434,TEST]=*.TST (RET)
FILES RENAMED:
DSKC:NUMB.TST
DSKC:109.TST
DSKC:FILL.TST
DSKC:PAY3.TST
```

Show that your UFD no longer contains the files with the extension .TST.

```
.DIR *.TST (RET)

%WLDNSF NO SUCH FILES AS DSKC:*.TST[27,5434]
```

Show that the files have been transferred to the SFD [27,5434,TEST].

```
.DIR [27,5434,TEST] (RET)

NUMB  TST  0  <055>  27-DEC-80      DSKC: [27,5434,TEST]
109   TST  5  <055>  20-NOV-80
FILL  TST  1  <055>  30-NOV-80
PAY3  TST  1  <055>  12-DEC-80
      TOTAL OF 7 BLOCKS IN 4 FILES ON DSKC: [27,5434,TEST]
```

2. The following example shows the use of CREDIR with the SCAN switch /PROTECTION.

```
.R CREDIR (RET)

CREATE DIRECTORY:  DSKB:[, ,A,B,C]/PROTECTION:755
                  CREATED DSKB:[27,4072,A].SFD/PROTECTION:755
                  CREATED DSKB:[27,4072,A,B].SFD/PROTECTION:755
                  CREATED DSKB:[27,4072,A,B,C].SFD/PROTECTION:755
CREATE DIRECTORY: ^C
```

## CREF Command

### Function

The CREF command runs the CREF program. If you have created any files to be processed with CREF (using the /CREF switch with a COMPILE, LOAD, DEBUG, or EXECUTE command), CREF processes them and prints them on the line printer. The file containing the names of these CREF files is then deleted so that subsequent CREF commands will not list them again.

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 file name and the extension .LST. (See the CREF Manual in the TOPS-10 Software Notebooks for more information.)

### Formats

CREF file-spec

Where: **file-spec** is a valid file specification. (Refer to Section 1.8.) When you supply a file specification in the command line, CREF produces a cross-referenced listing file for the specified file. If you do not give an argument to the command, CREF uses the argument saved from a previous COMPILE-class command. If there is no stored argument, CREF prompts with an asterisk (\*).

### Characteristics

The CREF command:

Leaves your terminal at monitor level.

Destroys your core image.

Requires LOGIN.

### Examples

1. Compile the files contained in the command file PROMAC and produce CREF listing files on the disk.

```
.COMPILE/CREF@PROMAC RET
FORTRAN: INPUT1
MAIN.
FORTRAN: INPUT2
MAIN.
```

Process and list the cross-referenced listing files produced by the COMPILE command. The argument is the stored argument that was used in the COMPILE command.

```
.CREF RET
```

SYSTEM COMMANDS AND PROGRAMS

**CREF Command (Cont.)**

2. Compile and load the files contained in the command file CONALL. Produce a loader map with the file name NAME and CREF files on disk.

```
.LOAD/CREF/MAP:NAME@CONALL (RET)  
MACRO: HIGH  
MACRO: SHARE  
EXIT
```

Assign the logical name LPT to magnetic tape unit 1. Store the CREF files on MTA1: to be output at a later time.

```
.ASSIGN MTA1 LPT (RET)  
MTA261 ASSIGNED
```

```
.CREF (RET)  
.
```

## CSTART Command

### Function

The CSTART command begins executing the program from the beginning, or from the address you specify in the command, leaving your terminal at monitor level. The CSTART command is the same as the START command, except that it leaves your terminal at monitor level.

### Format

CSTART addr

Where: **addr** is the address at which execution is to begin if it is other than the location specified within the file (.JBSA). If you do not specify an address, the starting address comes from .JBSA (stored in the job data area).

To use CSTART:

1. Use LOAD or GET to bring a program into memory.
2. Type one or two CTRL/Cs to halt your job with your terminal at monitor level.
3. Type CSTART to begin running the program from the beginning.
4. You can now type additional commands from your terminal if they do not require core, or you can detach your terminal from the job, using the DETACH command.

### Characteristics

The CSTART commands

Leaves your terminal at monitor level.

Requires core.

### Restrictions

This command should not be used if your program requests input from the terminal. This command is not available to batch users.

### NOTE

The CSTART command allows the program to run from the beginning or from the specified address. If the program requires terminal I/O, the program will wait until you access user level with the START or CONTINUE commands. Then your terminal can accept output or provide input.

## CSTART Command (Cont.)

## Example

This is a program to find all the numbers up to 10,000.

```
.TYPE NUMBER.FOR 
      N = 0
100   N = N + 1
      IF(N .EQ. 10000) GO TO 300
      WRITE(22,201) N
      GO TO 100
201   FORMAT(1X, I14, 'IS BETWEEN 1 AND 10000')
300   STOP 'DONE'
      END
```

Execute the program.

```
.EXECUTE NUMBER.FOR 
FORTRAN:NUMBER
NUMBER
LINK:LOADING
[LNKXCT NUMBER EXECUTION]
^C
^C
```

Type two CTRL/C's to halt the program.

Type CSTART.

```
.CSTART 
```

Type CTRL/T for job status. CTRL/T does not echo on your terminal.

```
DAY: :07:20 RUN: :01:00 RD:1093 WR:21 NUMBER 4+15P RN* PC000200
```

The status message RN\* indicates the NUMBER program is running.

Detach from the job to do work on another job.

```
.DETACH 
FROM JOB 19
```

.  
.  
.

Later, attach to your original job.

```
.ATTACH 19 [27,5434] 
PASSWORD: 
```

**CSTART Command (Cont.)**

Type CONTINUE to return to user level. System message (see NOTE).

.CONTINUE

?PLEASE TYPE ^C FIRST

Type CTRL/C and CONTINUE, to access user level.

.^C

.CONTINUE

DONE

END OF EXECUTION

CPU TIME:4:6.88 ELAPSED TIME:5:30.97

EXIT

Program message indicates execution is finished. The file containing data from this program is named FOR22.DAT, and is stored in your default disk area.

NOTE

It is necessary to type CTRL/C to re-enter user level, because the program is running. The program must be interrupted so that your terminal can access it.

**D(posit) Command****Function**

The D command deposits information in your core area (high or low segment). This function is useful for debugging programs.

DAEMON will be called on execution of the D command, if the page in question is paged out.

**Format**

D lh rh addr

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

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

addr is the address of the location into which the information is to be deposited. This argument is optional. If you omit the address, the system deposits the data into the location following the address you specified in the last D command or into the location you looked at with your last E command. The command used is the one specified most recently.

**Characteristics**

The D command:

Leaves your terminal at monitor level.

Requires core.

**Example**

Deposit in location 141.

```
.D 266000 2616 141 RET
```

Examine location 140.

```
.E 140 RET
000140/ 265720 40011
```

Because the address is omitted, the deposit made is in the location of the last E command (that is, location 140).

```
.D 47000 1 RET
```

Examine the location specified in the previous D command.

```
.E RET
000140/ 047000 000001
.
```

## DAYTIME Command

### Function

The DAYTIME command prints the date and the time of day, in the following format:

```
dd-mmm-yy hh:mm:ss
```

Where:    dd is day  
          mmm is month  
          yy is year  
          hh is hours  
          mm is minutes  
          ss is seconds to the nearest hundredth

### Format

```
DAYTIME
```

### Characteristics

The DAYTIME command:

Leaves your terminal at monitor level.

Does not require LOGIN.

Does not destroy your core image.

### Example

```
.DAYTIME RET  
17-JAN-80 11:18:11
```

.



## DCORE Command

### Function

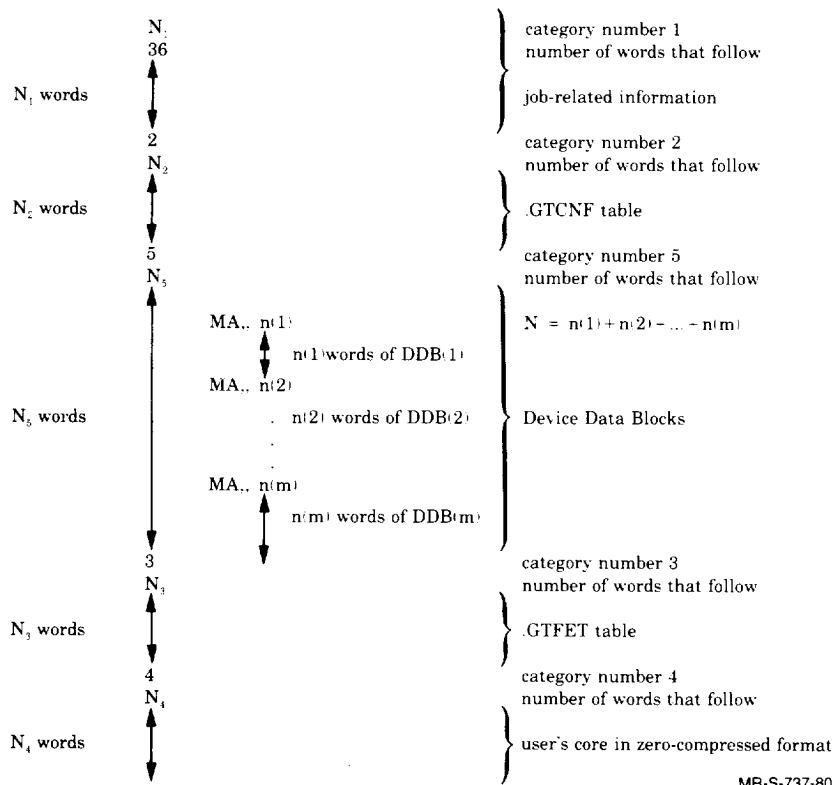
The DCORE command causes the DAEMON program to write a file of your core-image. The image file includes all accumulators and all relevant job information. The job can continue to run; the DCORE command does not destroy your core image. You can use the DUMP program to process the image file later.

The DAEMON-written file consists of five categories: JOB, CONFIGURATION, DDB, CORE, and FEATURES. Each category has a 2-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:

Mnemonic	Category Number	Description
JOB	1	Job-related information.
CONFIGURATION	2	The Configuration Table (.GTCNF) from the GETTAB monitor call.
DDB	3	The device data blocks (DDB) assigned to this job.
CORE	4	Your core area, both the low and the high segments, in zero-compressed format. (This format is described in the <u>TOPS-10 Monitor Calls Manual</u> .)
FEATURES	5	The Feature Table (.GTFET) from the GETTAB monitor call.

The third word of each category is the beginning of the data from that category. The DAEMON-written file is in the format shown in Figure 2-1.

DCORE Command (Cont.)



MR-S-737-80

Figure 2-1 DAEMON-Written File

SYSTEM COMMANDS AND PROGRAMS

DCORE Command (Cont.)

Categories can be written in any order. Version 17 of DAEMON writes the categories in the following order: 1, 2, 5, 3, 4, but your programs should not assume any special order.

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

Word	Description
0/	DAEMON's version number
1/	Today's date (from the DATE monitor call)
2/	Time of day in milliseconds (from the MTIME monitor call)
3/	Left half: job-number Right half: segment number
	These quantities are used as indexes into the GETTAB tables for items 5 through 52.
4/	Left half: reserved (not used) Right half: line number

The remaining entries are from the GETTAB tables. In some cases, the job-number is used as the index; in other cases, the segment number is used at the index. This is indicated as "J" or "S" in the third column below.

Table 2-1  
Job Information Data Words 5 through 52

Word	GETTAB Table	Job/Segment	Description
5	.GTSTS	J	Job Status
6	.GTSTS	S	Job Status
7	.GTPPN	J	Project-programmer number
10	.GTPPN	S	Project-programmer number
11	.GTPRG	J	Program name
12	.GTPRG	S	High-segment name
13	.GTTIM	J	Total runtime in jiffies
14	.GTKCT	J	Kilo-core ticks used by job
15	.GTPRV	J	Privilege bits
16	.GTSWP	J	Swapping parameters

SYSTEM COMMANDS AND PROGRAMS

DCORE Command (Cont.)

Table 2-1 (Cont.)  
Job Information Data Words 5 through 52

Word	GETTAB Table	Job/Segment	Description
17	.GTSWP	S	Swapping parameters
20	.GTRCT	J	Disk blocks read by job
21	.GTWCT	J	Disk blocks written by job
22	.GTTDB	J	Obsolete
23	.GTDEV	S	Device name of sharable high segment
24	.GTNM1	J	User name in SIXBIT (first half)
25	.GTNM2	J	User name in SIXBIT (second half)
26	.GTCNO	J	Charge number for job
27	.GTIMP	J	Obsolete
30	.GTWCH	J	WATCH bits
31	.GTSPL	J	Spooling control bits
32	.GTRTD	J	Real-time status word
33	.GTLIM	J	Time limit and batch status
34	.GTSPS	J	Job status word - 2nd processor
35	.GTRSP	J	Response counter table
36	.GTTRQ	J	Total elapsed time in run queue
37	.GTUPM	J	Physical page number of user page map
40	.GTUPM	S	Physical page number of user page map

SYSTEM COMMANDS AND PROGRAMS

DCORE Command (Cont.)

Table 2-1 (Cont.)  
Job Information Data Words 5 through 52

Word	GETTAB Table	Job/Segment	Description
41	.GTCVL	J	Left half: virtual page limit Right half: physical page limit
42	.GTMVL	J	Left half: maximum virtual page limit Right half: maximum physical page limit
43	.GTIPA	J	IPCF statistics
44	.GTIPP	J	Left half: pointer to next IPCF message in receive queue: Bits 18-26: number of messages sent but not retrieved by receiver Bits 27-35: number of IPCF messages in receive queue
45	.GTIPI	J	PID for job's local INFO
46	.GTIPQ	J	IPCF flags and quotes
47	.GTDVL	J	Pointer to logical name table
50	.GTABS	J	Address-break word
51	.GTVRT	J	Paging rate for job

Category 2 contains the entries in the .GTCNF GETTAB table. (See the TOPS-10 Monitor Calls Manual.)

Category 3 contains the device data blocks (DDBs) currently being used for this job. Each DDB is preceded by a word containing the monitor address in the left half, and the DDB length in the right half.

Category 4 is a compressed core image of both the high and low segments. It contains only nonzero words.

Category 5 lists the feature test switches as obtained from the GETTAB Table .GTFET. (See the TOPS-10 Monitor Calls Manual.)

## DCORE Command (Cont.)

### Format

DCORE dev:file.ext[directory]

Where: **dev:** is a disk-like device on which the core image file is to be written. If you omit this argument, the system assumes DSK:.

**file.ext** is the name of the file to be written. The default file name is nnnDAE, where nnn is your job-number in decimal, and the default extension is .TMP. If you supply only the file name, the default file name extension is .DAE.

[**directory**] is a disk area other than your directory area. If you omit this argument, the system assumes that your disk area is to be used.

### Characteristics

The DCORE command:

Leaves your terminal at monitor level.

Does not destroy your core image.

### Restriction

DCORE is illegal when the program in memory is execute-only.

### Examples

1. The core image is written on your area of the disk with the name nnnDAE.TMP, where nnn is your job-number.

```
.DCORE (RET)
```

.

2. The core image file is written in your area on DSKB with the name FILEC.DAE.

```
.DCORE DSKB:FILEC (RET)
```

.

## DDT Command

### Function

The DDT command starts DDT, the dynamic system debugger. If DDT is already loaded with your core image, DDT starts at the address given by the right half of .JBDDT in the Job Data Area. (The Job Data Area stores information pertinent to your job. It is described in the TOPS-10 Monitor Calls Manual.) If DDT is not yet loaded, the monitor tries to merge a special version of DDT (called VMDDT) into the address space starting at location 700000.

If DDT is not yet loaded and the monitor cannot read in VMDDT, it prints the message: ?NO START ADDRESS. The monitor will not read in VMDDT if your core image is execute-only.

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 the monitor). DDT contains commands to allow you to start or resume at any desired address.

If your job was executing a monitor call when interrupted (at monitor level and not in TTY input wait or SLEEP mode), the monitor sets a status bit (UTRP) and continues the job at the location where it was interrupted. When the monitor call processing is complete, the monitor clears the status bit, sets .JBOPC to the address following the monitor call, and then traps to the DDT address found in .JBDDT.

If your job is at monitor level and in TTY INPUT WAIT or SLEEP mode, the trap to the DDT address occurs immediately and .JBOPC contains the address of the monitor call. If your job is at user level, 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.)

### Format

DDT

### Characteristics

The DDT command:

Places your terminal at user level.

If .JBDDT is zero, the monitor will merge SYS:VMDDT.EXE at location 700000.

## DDT Command (Cont.)

## Example

The following example shows how the DEBUG and DDT commands are used. Begin by writing a simple program with a spelling (syntactical) error.

```
.TYPE TEST.MAC 
  TITLE TEST SIMPLE PROGRAM
  SEARCH MACTEN,UOSYM

HIMSG: ASCIZ \HI THERE - THIS IS A SIMPLE TEST PROGRAM!
\
TEST: JCLF
  RESET
  OUTSTR HIMSG
  MONRT.

  END TEST
```

Use DEBUG to compile and load the program:

```
.DEBUG TEST.MAC 
MACRO: TEST
LINK: LOADING
?LNKUGS 1 UNDEFINED GLOBAL SYMBOL
  JCLF 0
[LNKDEB DDT EXECUTION]
DDT
^Z
```

Loading showed that the program contained an error, which is then corrected by using the SOS editor to replace "JCLF" with the correct "JFCL".

```
.SOS TEST.MAC 
EDIT: TEST.MAC

*SJCLF JFCL .  
600 TEST: JFCL
*ES 

[DSKC:TEST.MAC]
```

After correcting the file, reload the program, again using DEBUG.

```
.DEBUG TEST.MAC 
MACRO: TEST
LINK: LOADING
[LNKDEB DDT EXECUTION]
DDT
^Z
```



## DDT Command (Cont.)

The debugger successfully loaded the program; no errors were detected. The new TEST.REL file automatically supersedes the old .REL file. Next, save the core image of the loaded program, and begin execution.

```
.SAVE TEST 
TEST SAVED
```

```
.START 
```

```
HI THERE - THIS IS A SIMPLE TEST PROGRAM!
```

The program ran successfully. However, when you try to run the program again with the CONTINUE command, the command fails. No provision has been made for rerunning the program.

```
.CONTINUE 
```

```
?ILLEGAL UO AT USER PC 013303
```

Start the debugger, using the DDT command:

```
.DDT 
DDT
```

```
13303/ 0 ^
TEST+3/ MONRT. 
PAT../ 0 JRST TEST 
^Z
```

While running DDT, alter the core image of the program: since the error was reported at address 13303 (octal), examine that address. Then type an uparrow (^) to see the previous line, which is the last line of program code. Then type a line-feed (<LF>) to see the next line. "PAT.." is on the next line. Here, insert the call "JRST TEST," then exit DDT.

Next, start the program:

```
.START 
```

```
HI THERE - THIS IS A SIMPLE TEST PROGRAM!
```

Then CONTINUE the program. The execution is successful.

```
.CONTINUE 
```

```
HI THERE - THIS IS A SIMPLE TEST PROGRAM!
```

## DDT Command (Cont.)

Finally, edit the file with the changes made to the core image:

```
.SOS TEST.MAC(RET)
EDIT:TEST.MAC
*p^:(RET)
100 TITLE TEST SIMPLE PROGRAM
200 SEARCH MACTEN,UOSYM
300
400 HIMSG: ASCIZ \HI THERE - THIS IS A SIMPLE TEST PROGRAM!
500 \
600 TEST: JFCL
700 RESET
800 OUTSTR HIMSG
900 MONRT.
1000
1100 END TEST
*I950(RET)
950 JRST TEST
*ES(RET)
[DSKC:TEST.MAC]
```

Load the program, begin execution, and rerun the program:

```
.LOAD TEST.MAC(RET)
LINK: LOADING
EXIT

.START(RET)

HI THERE - THIS IS A SIMPLE TEST PROGRAM!

.CONTINUE(RET)

HI THERE - THIS IS A SIMPLE TEST PROGRAM!

.
```

Execution is successful.

## DEASSIGN Command

### Function

The DEASSIGN command returns one or more devices currently assigned to your job to the monitor's pool of available devices and clears the logical names associated with them. If you are running an assembly-language program, and have an INITIALIZED device, it is not returned to the system's pool unless you have issued a RELEASE or RESET monitor call; only the device's logical name is cleared.

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. (See the TOPS-10 Monitor Calls Manual for further details.)

### Format

DEASSIGN dev:

Where: **dev:** is either the logical or the physical device name. This argument is optional. If you do not specify this argument, the system deassigns all devices from your job except your job's controlling terminal. Also, the system clears any logical name that might be associated with the controlling terminal.

### Associated Messages

If you specify a nonexistent device, the monitor prints:

?NO SUCH DEVICE

and leaves all current device assignments. If you specify a device that has never been ASSIGNED to your job, the monitor prints:

?devxxn WASN'T ASSIGNED

and leaves all current job assignments.

### Characteristics

The DEASSIGN command:

Leaves your terminal at monitor level.

Does not destroy your core image.

Requires LOGIN.

## DEASSIGN Command (Cont.)

## Example

Assign a card reader to your job. Card reader number 1 is assigned.

```
.ASSIGN CDR:RET
CDR261 ASSIGNED
```

Card reader number 1 is busy because it is assigned to your job. Show the devices that are busy.

```
.SYSTAT BRET
```

```
BUSY DEVICES:
DEVICE      JOB      WHY      LOGICAL

TTY144      18      INIT
TTY235      18      INIT
LPT260      18      INIT
LPT261      18      INIT
LPT263      18      INIT
CDR260      18      INIT
CDR261      24      AS
```

Deassign the card reader.

```
.DEASSIGN CDRRET
```

Again, show the busy devices.

```
.SYSTAT BRET
```

```
BUSY DEVICES:
DEVICE      JOB      WHY      LOGICAL

TTY144      18      INIT
TTY235      18      INIT
LPT260      18      INIT
LPT261      18      INIT
LPT263      18      INIT
CDR260      18      INIT
```

Card reader 1 is not in the list of busy devices. Therefore, it is available to any user.

## DEBUG Command

### Function

The DEBUG command translates the specified source files, if necessary, loads the .REL files generated, and prepares the loaded program for debugging. A system debugging program is loaded first, followed by your program, including local symbols. Upon completion of loading, the system transfers control to the debugging program. The debugging program that is used depends on the first file in the command string. If this file is a COBOL source file, COBDDT (the COBOL debugging program) is used. If the file is a FORTRAN source file, FORDDT is used.

When the first file contains a program that was previously compiled (that is, with extension .REL) and you want to use COBDDT as the debugging program, the /COBOL switch must appear in the command string. For example:

```
.DEBUG FILEA.REL/COBOL
```

Generally, a program debugged with the DEBUG command requires more core to execute than the same program compiled with the EXECUTE command requires. Extra space is occupied by the debugging program and additional debugging information, such as local symbols.

Each time the system executes a COMPILER, LOAD, EXECUTE, or DEBUG command, the system stores the command argument in a temporary file. When you issue one of these commands without arguments, the system uses the arguments stored in the temporary file. (Refer to Appendix C.) This command runs the COMPIL program before it runs the appropriate compiler and debugger.

### Format

```
DEBUG file-spec
```

Where: **file-spec** is a single file specification or a string of file specifications, separated by commas. A file specification consists of a device name, a file name (with or without an extension), and a directory name. (Refer to Section 1.8.)

The following switches can be used to modify the command string:

Switch	Function
/ALGOL	Compiles the file with ALGOL. Assumed for files with the extension of .ALG.
/BIN	Generates a binary file for each file compiled. The file extension of the output file is .REL. This is the default action.
/BLISS	Compiles the file with BLISS10. Assumed for files with the extension of .Bl0 and .BLI.
/C68 /C74	Runs the specified COBOL version.

## SYSTEM COMMANDS AND PROGRAMS

### DEBUG Command (Cont.)

Switch	Function
/COBOL	Compiles the file with COBOL. Assumed for files with the extension of .CBL.
/COMPILE	Forces a compilation of this file even if a binary file exists with a newer date and time than the source file. This switch causes an extra compilation, because compilation is not normally performed if the binary file is newer than the source file.
/CREF	Produces a cross-referenced listing file on the disk for each file compiled for later processing by the CREF program. The file extension of the output file 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. You must issue the CREF command to obtain the listing.
/DDT	Loads DDT and disregards the extension of the first file in the command string. This switch applies to all subsequent files.
/DEBUG: (arg, arg,...)	Passes the specified arguments to FORTRAN. Refer to the <u>FORTRAN-10 Programmer's Reference Manual</u> .
/DLIST	Creates a .LST file in your disk area. You can list the file on the line printer with the PRINT command.
/F10 /F40	Obsolete
/FOROTS /FORSE	Obsolete
/FORTRAN	Compiles the file with a FORTRAN compiler. Assumed for files with the extension .F4 and .FOR and all files with unrecognizable compiler extensions, if FORTRAN is the standard compiler. This switch is necessary if the file has a unrecognizable compiler extension and FORTRAN is not the standard compiler or is not the current default.
/FUDGE:file	Creates a disk file containing the names of the .REL files produced by the command string. When the FUDGE switch is given, PIP reads this file to generate a library .REL file. (Refer to the FUDGE command description.) The argument to this switch is a valid file specification, as described in Section 1.8.

## SYSTEM COMMANDS AND PROGRAMS

### DEBUG Command (Cont.)

Switch	Function
/K?10	Designates the processor where the program will execute after it has been loaded. These switches are necessary for FORTRAN-10 programs because the compiler generates different codes for different processors. The default is the processor where your program is running. The ? can be replaced by A, I, or L.
/LIBRARY	Loads the file in library search mode. This mode causes a program file in a special library 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. (See the LINK documentation.)
/LINK	Obsolete
/LIST	Generates a disk listing file for each file compiled. The file extension of the output file is .LST. These files can be listed later with the PRINT command. If the line printer is being spooled for this job, the listing files are written on device LPT: and are automatically spooled when you log out.
/LMAP	Produces a loader map during the loading process that contains the local symbols.
/LOADER	Obsolete
/MACRO	Assembles the file with MACRO. Assumed for files with the .MAC extension.
/MACY11	Assembles the file with MACY11. Assumed for files with the .P11 extension. This switch is not supported.
/MANTIS	Compiles the file with the MANTIS debugging information. This switch affects FORTRAN-40 programs only. This switch is not supported.
/MAP:file	Produces a loader map during loading. The file name can be specified. If the file is not specified, the default is MAP.MAP.
/NEW	Runs the appropriate language compiler from the experimental system library (device NEW:) area [1,5]. If the compiler does not exist on device NEW:, COMPIL tries to obtain it from device SYS:.

## SYSTEM COMMANDS AND PROGRAMS

### DEBUG Command (Cont.)

Switch	Function
/NOBIN	Does not generate binary files. Binary files are generated unless you give this switch. This switch, when combined with the /LIST or /CREF switch, is useful when compiling programs solely for the purpose of generating a listing.
/NOCOMPILE	Complement to the /COMPILE switch, this switch does not force a compilation on a source file whose date is not as recent as the date on the binary file. This switch differs from the /REL switch, in that it turns off all compilation, even if the source file is newer than the .REL file. /NOCOMPILE is the default action.
/NODEBUG	Does not pass arguments that were previously specified to FORTRAN.
/NOLIST	Does not generate listing files. This is the default action.
/NOMANTIS	Compiles the file without the MANTIS debugging information. This switch affects FORTRAN-40 programs only. This switch is not supported.
/NOOPTIMIZE	Does not optimize the object source code for FORTRAN programs.
/NOSEARCH	Loads all routines of the file whether the routines are referenced or not. Because this is the default action, this switch is used only to turn off library search mode (/LIBRARY). The /NOSEARCH default is to search the system libraries.
/OLD	Runs the appropriate language compiler from the system library of old programs (device OLD:), which resides on disk area [1,3]. If the compiler does not exist on device OLD:, COMPIL tries to obtain it from device SYS:.
/OPTIMIZE	Optimizes the object source code for FORTRAN programs.
/REL	Uses the existing .REL files although newer source files might be present.
/SAVE	Saves the core image of the loaded program.



## SYSTEM COMMANDS AND PROGRAMS

### DEBUG Command (Cont.)

Switch	Function
/SEARCH	Loads the files in library search mode. The action is identical to that of the /LIBRARY switch.
/SELF	Runs the appropriate language compiler from device DSK: instead of from the system library (device SYS:). This switch is useful if you keep a private copy of a compiler in your own disk area.
/SNOBOL	Compiles the file with SNOBOL. Assumed for files with the .SNO extension. This switch is not supported.
/SSAVE	Saves the core image of the loaded program in a sharable executable file.
/SYS	Runs the appropriate language processor from the system library (device SYS:) area of [1,4]. This is the default action.

#### Restriction

A language processor appearing more than once within a single command string cannot specify more than one disk area. For example, the following is invalid:

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

However, the following is valid:

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

#### Characteristics

The DEBUG command:

Places your terminal at user level.

Runs the appropriate processor, LINK, and the debugger, destroying your core image.

## DEBUG Command (Cont.)

## Example

The following example shows how the DEBUG and DDT commands are used. Begin by writing a simple program with a spelling (syntactical) error.

```
.TYPE TEST.MAC 
  TITLE TEST SIMPLE PROGRAM
  SEARCH MACTEN,UOSYM

HIMSG:      ASCIZ  \HI THERE - THIS IS A SIMPLE TEST PROGRAM!
\
TEST: JCLF
      RESET
      OUTSTR HIMSG
      MONRT.

      END TEST
```

Use DEBUG to compile and load the program:

```
.DEBUG TEST.MAC 
MACRO: TEST
LINK: LOADING
?LNKUGS 1 UNDEFINED GLOBAL SYMBOL
      JCLF 0
[LNKDEB DDT EXECUTION]
DDT
^Z
```

Loading showed that the program contained an error, which is then corrected by using the SOS editor to replace "JCLF" with the correct "JFCL".

```
.SOS TEST.MAC 
EDIT: TEST.MAC

*SJCLF  JFCL  .  
600 TEST: JFCL
*ES

[DSKC:TEST.MAC]
```

After correcting the file, reload the program, again using DEBUG.

```
.DEBUG TEST.MAC 
MACRO: TEST
LINK: LOADING
[LNKDEB DDT EXECUTION]
DDT
^Z
```

**DEBUG Command (Cont.)**

The debugger successfully loaded the program; no errors were detected. The new TEST.REL file automatically supersedes the old .REL file. Next, save the core image of the loaded program, and begin execution.

```
.SAVE TEST(RET)
TEST SAVED
```

```
.START(RET)
```

```
HI THERE - THIS IS A SIMPLE TEST PROGRAM!
```

The program ran successfully. However, when you try to run the program again with the CONTINUE command, the command fails. No provision has been made for rerunning the program.

```
.CONTINUE(RET)
```

```
?ILLEGAL UOO AT USER PC 013303
```

Start the debugger, using the DDT command:

```
.DDT(RET)
DDT
```

```
13303/      0  ^
TEST+3/    MONRT.(LF)
PAT../      0      JRST TEST(RET)
^Z
```

While running DDT, alter the core image of the program: since the error was reported at address 13303 (octal), examine that address. Then type an uparrow (^) to see the previous line, which is the last line of program code. Then type a line-feed (<LF>) to see the next line. "PAT.." is on the next line. Here, insert the call "JRST TEST," then exit DDT.

Next, START the program:

```
.START(RET)
```

```
HI THERE - THIS IS A SIMPLE TEST PROGRAM!
```

Then CONTINUE the program. The execution is successful.

```
.CONTINUE(RET)
```

```
HI THERE - THIS IS A SIMPLE TEST PROGRAM!
```

SYSTEM COMMANDS AND PROGRAMS

DEBUG Command (Cont.)

Finally, edit the file with the changes made to the core image:

```
.SOS TEST.MAC (RET)
EDIT:TEST.MAC
*P^:(RET)
100 TITLE TEST SIMPLE PROGRAM
200 SEARCH MACTEN,UOSYM
300
400 HIMSG: ASCIZ \HI THERE - THIS IS A SIMPLE TEST PROGRAM!
500 \
600 TEST: JFCL
700 RESET
800 OUTSTR HIMSG
900 MONRT.
1000
1100 END TEST
*I950 (RET)
950 JRST TEST
*ES (RET)
[DSKC:TEST.MAC]
```

Load the program, begin execution, and run the program again:

```
.LOAD TEST.MAC (RET)
LINK: LOADING
EXIT

.START (RET)

HI THERE - THIS IS A SIMPLE TEST PROGRAM!

.CONTINUE (RET)

HI THERE - THIS IS A SIMPLE TEST PROGRAM!
```

Execution is successful.

## DELETE Command

### Function

The DELETE command deletes files. Because of protection codes associated with files, it is not usually possible to delete files in another user's directory. Once a file is deleted, it cannot be recovered.

### Format

DELETE file-spec

Where: **file-spec** is a single file specification or a string of file specifications, separated by commas. The full wildcard construction (\* and ?) can be used in the file specification. (Refer to Appendix B.)

If you do not specify a device name or a file structure name, your job's search list is used. You can specify a directory name before the file names and that directory becomes the default for all subsequent files in that command line. If you specify a directory name after a file name, the directory applies only to that file.

### Characteristics

The DELETE command:

Leaves your terminal at monitor level.

Runs the PIP program.

Requires LOGIN.

Permanently deletes the file.

Destroys your core image.

### Examples

1. Delete all files with .MAC extension.

```
.DELETE *.MAC 
```

```
FILES DELETED:  
DSKB:T1.MAC  
DSKB:T2.MAC  
DSKB:T3.MAC  
14 BLOCKS FREED
```

**DELETE Command (Cont.)**

2. Delete the file TEST.FOR.

```
.DELETE DSKC:TEST.FOR 
```

```
FILES DELETED:  
DSKC:TEST.FOR  
3 BLOCKS FREED
```

.

3. Delete all files with the file name TEST followed by 2 alphanumeric characters or less, and the extension .FOR.

```
.DELETE TEST??.FOR 
```

```
FILES DELETED:  
DSKB:TEST1.FOR  
DSKB:TEST2.FOR  
DSKB:TEST22.FOR  
DSKB:TESTER.FOR  
23 BLOCKS FREED
```

.

## DETACH Command

### Function

The DETACH command disconnects the terminal from your job without changing the status of your job. Your terminal is then free, so that you can start a new job, attach to another detached job, or cause your terminal to become a slave terminal for another job.

### Format

DETACH

### Characteristics

The DETACH command:

Detaches your terminal.

Does not destroy your core image.

Requires LOGIN.

### Restrictions

This command is not available to batch users.

### Example

Show the status of your job.

```
.SYSTAT .RET  
1 27,5434 TTY263 SYSTAT 6+7 ^C
```

System message shows that your terminal line number is 263.  
Detach your terminal.

```
.DETACHRET  
FROM JOB 24
```

Show the status of your job.

```
.SYSTAT .RET  
1 27,5434 DET SYSTAT 6+7 ^C
```

System message shows your job is detached.

**DIRECTORY Command****Function**

The DIRECTORY command prints a list of the file names in a specified directory area. The standard output consists of the following information: file name, file name extension, length of the file in blocks, protection code, creation date, version number, account, structure name, and directory name.

**Format**

DIRECTORY output file-spec=input file-spec

Where: **input file-spec** is 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:, or MTA: 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 BACKUP tape. The default input specification is DSK:\*.\*, and the files in all file structures defined by your job's search list are listed. Generally, a device name, an extension, or a directory name that precedes the file name becomes the default for all succeeding files in the list.

**output file-spec** = This argument and the equal sign are optional. If you omit the entire output specification, the default is TTY:. If you do not specify an output device, the default device is DSK:. If you do not specify an output file name, and one is needed, the file name is generated from the time of day as hhmmss. The default output extension is .DIR.

The full wildcard construction (\* and/or ?) can be used in the input file-spec. When a wildcard designation is used, the DIRECTORY program limits its search for the file to certain directories. When you give a wildcard designation for a file name or extension, the program only searches the specified directory or your default directory. No additional devices, such as LIB: or SYS:, which might be in your default path, are searched. (Refer to Section 1.13, and to the PATH monitor call description in the TOPS-10 Monitor Calls Manual, for information about directory paths.)

If you use the wildcard construction in the directory name, only the directories implied by the wildcard construction are searched. No additional directories are searched.

The following switches can be used in the command string. Generally, any switches can be used together in the same command string, unless the switches contradict one another. Switches that precede the file name become the default for all succeeding files in the same command line.

You can abbreviate switches as long as the result is unique. This is not recommended for batch control files. Spaces are not permitted within a switch.



## SYSTEM COMMANDS AND PROGRAMS

### DIRECTORY Command (Cont.)

Switch	Function
/ACCESS:n	Updates the access date of any file of n blocks or less to the current date. Because some installations delete files that have not been recently accessed, this switch allows you to prevent such deletion by updating the files. n is a decimal number, and refers to the number of blocks written in the file unless the /ALLOC switch is also used. If you omit the /ACCESS switch, the system does not change the date. If you specify /ACCESS but you omit :n, 5 is assumed.
/ACCOUNT	Prints the account name associated with that file. The account you are logged in under is stored in the RIB of all files created or superseded.
/ALLOCATED	Lists 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 can be greater than the number of blocks actually written. The LOGOUT program uses the allocated length when checking quotas. The total allocated length of all files is the same as the length output by the QUOLST program under the USED column. Normally, when a file is created, the system allocates 30 blocks for it. Then the system deallocates unused blocks after file creation is complete. This switch is the complement to the /WRITTEN switch.
/AUTHOR	Prints the project-programmer number of the author of the file.
/BEFORE:date-time	Lists those files created before the specified date and time. Default is +infinity. Refer to Section 1.7.3.
/BLOCKS	Prints the length of the file in blocks. This is the default. Complement to /WORDS.
/CHECKSUM	Computes and prints an 18-bit checksum for each file. This checksum is computed by rotating the result to the left one bit before adding each word. Complement to /NOCHECKSUM.
/COMPARE	Suppresses headers and titles, as well as error messages in the output. This makes the output file suitable for comparison with another file (with FILCOM). Default is /NOCOMPARE.

## SYSTEM COMMANDS AND PROGRAMS

### DIRECTORY Command (Cont.)

Switch	Function
/DENSITY:n	Uses the specified density when reading a magnetic tape. n is 200, 556, 800, 1600, or 6250 bpi. The default depends on your installation. You can change the default with the SET DENSITY command.
/DETAIL	Prints all available information about a file. The information includes:  The full file specification for the file. The access date. The time and date of creation. The access protection code associated with the file. The data mode that the file is written in. The estimated length of the file. The blocks allocated for the file. The data block in the directory in which the file is located. The internal date and time of creation. The RIB block number.  All numbers that are followed by a decimal point are decimal values; all other numbers are octal. The project-programmer number associated with the file is printed only if it is not the same as that of the user who issued the DIRECTORY command.
/DIRECT	Provides ASCII-formatted output. This the default. Complement to /NODIRECT.
/EOTS	Stops at the logical end of tape (two consecutive tape marks) when reading a magnetic tape. This is the default. Complement to /NOEOTS.
/FAST	Lists short form of directory: file name, extension, structure name, and directory name. Abbreviated to /F. Complement to /NORMAL and /SLOW.
/FDTA	Gets the DECTape directory from core.
/FILES:n	Stops after n files when reading a magnetic tape. If you specify /FILES but you omit :n, 5 is assumed. When the system reaches the logical EOT, it will stop reading the tape.
/FIND	Looks for the Find Files for the directory listing rather than the devices. A Find File is the binary output of a directory listing, and is created with the /FNDBLD switch. /FIND looks for the Find File in SYS:FNDDAT.FDF or .FPF. You can use the /FNDDAT switch to specify the file name of the Find File.

## SYSTEM COMMANDS AND PROGRAMS

### DIRECTORY Command (Cont.)

Switch	Function
/FLSDIR	Prints each file's device and directory to the right of the first line of output for each directory. Complement to /NOFLSDIR, the default is /FLSDIR, unless /HSDIR or /WIDTH is specified.
/FNDBLD	Creates Find Files from the DIRECTORY output. A Find File is the binary output from a DIRECTORY listing. Find Files are useful for archiving and retrieving files on magnetic tape. The file name of the Find File is that specified in the output specification of the command format. If you do not specify the output file name, the default is SYS:FNDDAT.FDF or .FPF. The files can later be accessed with the /FIND switch. The complement to /FNDBLD is /NOFNDBLD. /NOFNDBLD is the default.
/FNDDAT:file	Specifies the file name of the Find Files to be read for the directory listing. A Find File is the binary output for a directory listing, and is created with the /FNDBLD switch. You must use the /FIND switch with the /FNDDAT switch.
/HSDIR	Prints the device and directory information of the file as a separate header line, immediately preceding the directory listing for each directory. Complement to /NOHSDIR, the default is /NOHSDIR, unless /WIDTH is specified.
/HELP	Prints the HELP text that lists the switches available and how to use them. Abbreviated to /H.
/HELP:S	Lists all switches without their explanations. An asterisk prefixes those switches that have a single-letter abbreviation.
/INDIRECT	Creates a control file that allows you to sort directories or files.
/LIST	Queues the output to device LPT:. Abbreviated to /L. Refer to the LIST command for restrictions on this switch.
/MARKS	Indicates each tape mark and UFD when reading a magnetic tape. Complement of /NOMARKS.
/MVOLUME	When reading BACKUP and DUMPER magnetic tapes, asks the user to mount another reel when the end of tape is encountered in the middle of a save set. Complement to /NOMVOLUME, the default is /NOMVOLUME.

## SYSTEM COMMANDS AND PROGRAMS

### DIRECTORY Command (Cont.)

Switch	Function
/NOAUTHOR	Does not print the project-programmer number of the author of the file. This is the default. Complement to /AUTHOR.
/NOCHECKSUM	Does not compute and print the checksum. This is the default. Complement of /CHECKSUM.
/NOCOMPARE	Prints the normal headers, titles, and error messages. Complement to /COMPARE, the default is /NOCOMPARE.
/NODETAIL	Does not print the words in the LOOKUP block. This is the default. Complement to /DETAIL.
/NODIRECT	Does not print the normal ASCII listing. Complement to /DIRECT, the default is /DIRECT.
/NOEOTS	Does not stop at the logical end of tape when reading a magnetic tape. Complement to /EOTS.
/NOFDTA	Does not get the DEctape directory from core. This is the default. Complement to /FDTA.
/NOFIND	Does not look for Find Files for the output. Complement to /FIND, the default is /NOFIND.
/NOFLSDIR	Does not print each file's device and directory to the right of the first line for each directory listed. Complement to /FLSDIR; the default is /FLSDIR, unless /HSDIR or /WIDTH is specified.
/NOFNDBLD	Does not make a Find File from the output. Complement to /FNDBLD; the default is /NOFNDBLD.
/NOHSDIR	Does not print the device and directory as a header for each directory listed. Complement to /HSDIR, the default is /NOHSDIR, unless /WIDTH is specified.
/NOINDIRECT	Does not format the directory listing so that it can be used as input to a program. Complement to /INDIRECT, the default is /NOINDIRECT.
/NOMARKS	Does not indicate each tape mark and UFD when reading a magnetic tape. This is the default. Complement to /MARKS.

## SYSTEM COMMANDS AND PROGRAMS

### DIRECTORY Command (Cont.)

Switch	Function
/NOMVOLUME	When using BACKUP or DUMPER, does not ask the user to mount another magnetic tape when the end of tape comes in the middle of a save set. Complement to /MVOLUME, the default is /NOMVOLUME.
/NOPRDEVICE	Does not print the device name. Complement to /PRDEVICE.
/NOPRDIRECTORY	Does not print the directory. Complement to /PRDIRECTORY.
/NORMAL	Prints the normal directory list. This list includes the file name, extension, length in blocks written, protection, creation date, structure name, nonzero version numbers, and directory name. Complement to /FAST and /SLOW. This is the default. Use this switch to override a /FAST or /SLOW in your SWITCH.INI file.
/NOPRVERSION	Suppresses printing the version number of the files. The normal listing prints the version number only if it is not zero. Complement to /PRVERSION.
/NORETRY	Disables automatic error retry when reading a file. Generates error messages for soft errors. Complement to /RETRY, the default is /RETRY.
/NOREWIND	Does not rewind the tape before and after reading a magnetic tape. Complement to /REWINDS.
/NOSORT	Does not produce a file suitable for sorting. This is the default. Complement to /SORT.
/NOSUMMARY	Does not use summary mode. This is the default. Complement to /SUMMARY.
/NOTITLE	Does not print page headers. This is the default for output to the terminal. Complement to /TITLES.
/NOUNITS	Does not list the name of the actual disk unit; instead, just lists the structure name. This is the default. Complement to /UNITS.
/OKNONE	Suppresses the error message if no files match the wildcard construction.

## SYSTEM COMMANDS AND PROGRAMS

### DIRECTORY Command (Cont.)

Switch	Function
/OPTION:name	Reads your option file (DSK:SWITCH.INI) to determine your 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. For example, if the file contains the line:  DIRECT:ALL/DETAIL  then you reference this line by typing the command:  DIRECT/OPTION:ALL  Refer to Appendix B for additional information.
/PARITY:ODD /PARITY:EVEN	Specifies the parity to be used when reading a magnetic tape. The default is :ODD.
/PHYSICAL	Ignores logical names. Refer to Section 1.8.1 for a description of logical names.
/PRDEVICE	Prints the name of the device for each file.
/PRDIRECTORY	Prints the name of the directory for each file.
/PROTECTION:nnn	Gives the output file the protection nnn (octal). Protection codes are discussed in Section 1.8.4.
/PRVERSION	Prints the version of each file. If you do not specify this switch, the default is to print the version only if it is not zero. The complement is /NOPRVERSION.
/RETRY	Enables automatic error retry when reading a file. Complement to /NORETRY, the default is /RETRY.
/REWIND	Rewinds the magnetic tape before and after reading it. This is the default. Complement to /NOREWINDS.
/RUN:file-spec	Runs the specified program when this command is finished.
/RUNOFFSET:n	Runs the program specified with /RUN with an offset of n. If you omit the switch, the default is 0; if you omit the value, the default is 1.
/SINCE:date-time	Lists only those files created after the specified date and time. The default is January 1, 1964. Refer to Section 1.7.3.

## SYSTEM COMMANDS AND PROGRAMS

### DIRECTORY Command (Cont.)

Switch	Function
/SLOW	Prints a full listing that includes the file name, extension, length in blocks written, protection, access date, creation time and date, structure name, and directory name. Equivalent to /S. Complement to /FAST and /NORMAL. (Disk and magnetic tape only.)
/SORT	Lists the file structure name and directory name for each file. The file structure name is output for every file if you do not specify a file structure name in the command string or if you specify generic DSK:. The wildcard construction is used in the directory name. Multiple spaces are output instead of TABs. Project-programmer numbers include leading zeros; the date is in ANSI format: 19721009 for Oct 9, 1972. Use this switch to prepare a file to be sorted by the SORT program. (See the COBOL documentation.) Complement to /NOSORT.
/SUMMARY	Prints only the summary line that indicates the total number of blocks and files. A /FAST/SUMMARY prints a /FAST listing followed by the summary.
/TITLE	Causes 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 to /NOTITLES.
/UNITS	Lists the name of the actual disk unit on which the files are stored instead of the file structure name. Complement to /NOUNITS.
/WIDTH:n	Prints several entries on a single line to make the output n columns wide. If /FAST is specified for output to a terminal, four file names appear per line. The default for n is 64 columns.
/WORDS	Prints the length of the file in words instead of blocks. Complement to /BLOCKS.
/WRITTEN	Prints the written length of the file rather than the allocated length. This is the default. Complement to /ALLOCATE.

#### Characteristics

The DIRECTORY command:

Leaves your terminal at monitor level.

Destroys your core image.

# SYSTEM COMMANDS AND PROGRAMS

## DIRECTORY Command (Cont.)

### Examples

1. List all files on DSKB:.

```
.DIRECT DSKB: (RET)
```

```
FILE   DAT      220 <055> 15-AUG-79 31(225) DSKB: [27,5055]  
SWITCH INI      10 <057> 10-MAY-79  
PROG   MAC      5 <055> 11-SEP-79
```

.

2. List all files with extension .MAC in all file structures in your job's search list.

```
.DIRECT *.MAC (RET)
```

```
PROG MAC 5 <057> 10-MAY-79 31(225) DSKB:[27,5055]  
FILE MAC 1 <055> 8-JUL-79 30(201) DSKC:[27,5055]
```

.

3. List the directory entry for the file TEST.F4 in user area [27,4072].

```
.DIR TEST.F4[27,4072] (RET)
```

```
TEST F4 6 <055> 10-JUN-79 31(225) DSKC:[27,4072]
```

.

4. List all files in sub-file directory WIZZER.SFD in the [7,2] area on BLKT:.

```
.DIR BLKT:[7,2,WIZZER] (RET)
```

```
ATTA   TST      1 <055> 19-SEP-79 BLKT:[7,2,WIZZER]  
NEW    ALG      4 <055> 10-AUG-79  
ALGOL  DOC     20 <055> 20-OCT-79
```

.

5. Show the switches to the DIRECT command.

```
.DIRECT/HELP:SWITCH (RET)
```

```
SWITCHES ARE: ACCESS, ACCOUN, ALLOCA, AUTHOR, BLOCKS, CHECKS, COMPAR,  
DETAIL, DIRECT, EOTS, *FAST, FDTA, FILES, FIND, FLSDIR,  
FNDBLD, FNDDAT, HDSDIR, INDIRE, *LIST, MARKS, MVOLUM, *NORMA,  
PRDEVI, PRDIRE, PRVERS, RETRY, REWIND, *SLOW, SORT, SUMMAR,  
TITLES, UNITS, *WIDTH, WORDS, WRITTE
```

```
STANDARD ONES: ABEFOR, ASINCE, BEFORE, DENSIT, ERNONE, ERPROT, ERSUPE,  
ESTIMA, EXIT, *HELP, LENGTH, MESSAG, NOOPTI, OKNONE, OKPROT,  
OKSUPE, OPTION, PARITY, PHYSIC, PROTEC, RUN, RUNCOR, RUNOFF,  
SINCE, STRS, TMPFIL, VERSIO
```

.



**DISABLE Command****Function**

The DISABLE command disables the POKE, SPY, and other privileged monitor calls that were enabled when you first logged in or when you used the ENABLE command. When you log in, your privileges, if any, are enabled by default.

**Format**

DISABLE

**Characteristics**

The DISABLE command:

Does not destroy your core image.

Leaves your terminal at monitor level.

Does not change privileges associated with being [1,2] or on the operator's terminal.

Requires LOGIN.

**Example**

Run FILDDT.

```
.R FILDDT (RET)
```

Look at memory area. Privileges are required to examine address 41.

```
FILE: /M (RET)
41/   JSR3600
C
```

The contents of 41 are shown, indicating your job has privileges enabled. Disable the privileges:

```
.DISABLE (RET)
```

Run FILDDT.

```
.R FILDDT (RET)
```

When you attempt to examine address 41, the contents are not revealed. 0 is printed.

```
FILE: /M (RET)
41/   0
^C
```

SYSTEM COMMANDS AND PROGRAMS

**DISABLE Command (Cont.)**

Enable your privileges.

.ENABLE

Again, examine memory with FILDDT. Your job's privileges have been reinstated.

.R FILDDT   
FILE: /M   
41/ JSR3600  
^C

## DISMOUNT Command

### Function

The DISMOUNT command returns devices to the monitor's pool of available resources, and removes file structures from your job's search list. Restricted devices are returned to the restricted pool and unrestricted devices to the unrestricted pool. Also, you can use the DISMOUNT command to reassign an unrestricted device to another job, or to retain control of the device you are DISMOUNTing.

You can continue the job as soon as you issue the DISMOUNT command even if operator action is required. If you want to wait for the operator to acknowledge the request before continuing with the job, you must specify the /WAIT switch. A batch job always waits until the DISMOUNT has been completed, unless the /NOWAIT switch has been specified.

To ensure validity of any tape error analysis, you should use MOUNT and DISMOUNT to acquire and release magnetic tape units. This provides the basis for all media-related error reporting.

### Format

DISMOUNT dev:/switches

Where: **dev:** is any previously ASSIGNED or MOUNTed device or file structure name. The name can be a physical or logical name (unless you use the /REASSIGN switch). The device argument is required except when you use /HELP, /CHECK, or /WAIT.

**/switches** are switches that you can use to further specify the request. Switches are processed in the order that you specify them. If there are any conflicting switches in the command string, the last one in the string is used. (An exception to this is the use of both the /WAIT and /NOWAIT switches within a single command string. If both switches are specified, a fatal error results.) If you specify switches that do not apply to the device type, the switches are ignored. For example, DISMOUNT DSKB:/REWIND. You can abbreviate switches to a unique set of characters.

Switch	Function
/CHECK	Checks and lists your job's pending DISMOUNT requests. This switch is intended to be used without other switches and without a device specification. If other switches or a device appear in the same command string with the /CHECK switch, they will be ignored.
/HELP	Prints a brief description of the DISMOUNT command and the list of switches that are valid with it. This switch is intended to be used without other switches and without a device specification. If other switches or a device appear in the command string with the /HELP switch, they will be ignored.

## SYSTEM COMMANDS AND PROGRAMS

### DISMOUNT Command (Cont.)

Switch	Function
/KEEP	Retains control of the device for further use. Keeps the specified device assigned to your job, even though it is currently dismounted. This switch is not intended to be used with file structures.

#### NOTE

If you specify /REWIND, /STAY, or /UNLOAD, you must specify the /KEEP switch. Otherwise, these switches will not function.

/PAUSE	Notifies you before queuing the DISMOUNT request to the operator. This allows you to cancel the request before it is completed.
/REASSIGN:n	Reassigns the device to job number n. The device must be a physical device name. Restricted devices cannot be reassigned.
/RELEASE	Releases control of the device. This is the default function.
/REMOVE	Removes the file structure from your search list. If there are no other jobs using the file structure and if it does not contain system files, the operator is notified that you are finished with the device and that it should be removed.

#### NOTE

R is a valid abbreviation for REMOVE. Therefore, if you want to use /REASSIGN, /RELEASE, or /REWIND, you must type at least the first three characters of those switch names.

/REWIND	Rewinds the tape. This switch is ignored if the tape has already been rewound or unloaded.
/STAY	Does not reposition the device. This is ignored for disk.
/UNLOAD	Unloads the tape. This is the default function for tapes.
/VID:"text"	Specifies a visual identification to be passed to the operator. The text can be up to 50 characters in length and can be delimited by single or double quotation marks. The quotation marks are included in the 50 characters.

**DISMOUNT Command (Cont.)**

Switch	Function
/WAIT	Waits for the operator to acknowledge the DISMOUNT request before continuing your job. This switch is ignored if operator action is not required to complete the DISMOUNT. This is the default for batch jobs. The /WAIT switch can be used without a device specification, and without other switches. Your job will then wait for all pending DISMOUNT requests to be completed.
/NOWAIT	Allows your job to continue without waiting for operator response. This is the default function.

**Characteristics**

The DISMOUNT command:

Leaves your job at monitor level.

Destroys your core image.

**Examples**

1. This DISMOUNT command ends the association between the logical name SCRATCH and the physical drive DTA0:, releases the drive DTA0: from this job, and queues a request to the operator to remove the DECTape from the drive. You refer to the drive by the logical name (SCRATCH); had you referred to the drive by the physical name (DTA0:), the command would have had the same effect.

```
.ASSIGN DTA SCRATCH (RET)
DTA0 ASSIGNED
```

```
.DISMOUNT SCRATCH (RET)
DTA0 DISMOUNTED
```

SYSTEM COMMANDS AND PROGRAMS

**DISMOUNT Command (Cont.)**

2. This DISMOUNT command causes the tape to be rewound, but does not queue a request to the operator. Because the command includes the /KEEP switch, the driver remains assigned to the job, and the logical name remains associated with the drive. The only effect, then, is to rewind the tape.

```
.ASSIGN MTA: FAILSA (RET)
MTA0 ASSIGNED
```

```
.DISMOUNT FAILSA/KEEP/REWIND (RET)
MTA0 DISMOUNTED
```

```
.SYSTAT (RET)
 7 10,470244 TTY111 SYSTAT 12+SPY RN 4 $
$ MEANS EXECUTE ONLY
```

```
BUSY DEVICES:
DEVICE JOB WHY LOGICAL
MTA0 7 AS FAILSA
```

3. This example shows that MTA0: has been assigned to job 7, and the logical name FAILSA is associated with MTA0:. Job 9 has no devices assigned to it. The DISMOUNT command releases MTA0: from job 7 and assigns it to job 9. The association between the logical name FAILSA and MTA0: is ended. The tape is unloaded.

```
.SYSTAT 7 9 (RET)
 7 10,470244 TTY111 SYSTAT 12+SPY RN 5 $
$ MEANS EXECUTE ONLY
```

```
BUSY DEVICES:
DEVICE JOB WHY LOGICAL
MTA0 7 AS FAILSA
```

```
9 10,470245 DET 3 ^C SW 2
```

```
.DISMOUNT FAILSA/REASSIGN:9 (RET)
MTA0 DISMOUNTED
```

```
.SYSTAT (RET)
 7 10,470244 TTY111 SYSTAT 12+SPY RN 6 $
$ MEANS EXECUTE ONLY
```

```
9 10,470245 DET 3 ^C SW 2
```

```
BUSY DEVICES:
DEVICE JOB WHY LOGICAL
MTA0 9 AS
```

## SYSTEM COMMANDS AND PROGRAMS

### DISMOUNT Command (Cont.)

4. In this example, you knew that the job was the only one using the structure BLKJ: and included a /REMOVE switch, so that the operator would remove the disk pack from the drive. The message "BLKJ DISMOUNTED" means that BLKJ: is no longer in your search list, and that a request has been queued to the operator to remove the pack from the drive. The message does not mean that BLKJ has already been physically removed. Because this command implies /NOWAIT, you will not receive notification of the physical removal of the disk pack.

```
.DISMOUNT BLKJ/REMOVE(RET)
BLKJ DISMOUNTED
```

The response to the following DISMOUNT/CHECK command tells you which of this job's dismount requests are still pending, and the total number of pending dismount requests in the system. BLKJ: has not yet been physically removed.

```
.DISMOUNT/CHECK(RET)
1. D JOB7 TTY111 10.470244 1 DISMOUNT BLKJ /R
1 COMMAND IN QUEUE
.
```

5. The following DISMOUNT/WAIT command causes the job to suspend further processing until all pending dismount requests from it have been completed. There were no pending dismount requests from this job, so the job returns to monitor level immediately.

```
.DISMOUNT/WAIT(RET)
NONE PENDING FOR YOUR JOB
.
```

6. In the following example, you request that DSKB: be removed from your search list and physically removed from the disk drive. DSKB: is removed from the job's search list, but because there are other jobs using DSKB:, the request to physically remove the pack is not queued to the operator.

```
.DISMOUNT DSKB/REMOVE(RET)
% DSKB HAS OTHER USERS - /REMOVE IGNORED
DSKB DISMOUNTED
```

```
.R SETSRC(RET)
```

```
*T(RET)
DSKA:/NOCREATE, DSKC:, DSKN:, FENCE
*^C
.
```

## DSK Command

### Function

The DSK command prints disk usage figures for all disk I/O performed since the last DSK command, followed by the total amount of disk I/O performed since the job was started. Disk usage is printed in the following format:

```
RD,WT=i,j  
RD,WT=m,n
```

Where: **i** and **j** are the incremental number of 128-word blocks read and written since the last DSK command.

**m** and **n** are the total number of 128-word blocks read and written since the job started.

### NOTE

**i** and **j** are kept modulo 4096. If you enabled automatic READ or WRITE output, using the SET WATCH command, **i** and **j** are usually zero, because SET WATCH resets these values. CTRL/T also resets these values.

### Format

DSK job

Where: **job** is the number of the job for which you want the disk usage. This argument is optional. If you include the job number in the command string, only the total statistics (**m** and **n** above) are printed.

If the job-number is omitted, the system prints the disk usage figures for your job.

### Characteristics

The DSK command:

Requires LOGIN.

Leaves your terminal at monitor level.

Does not destroy your core image.



**DSK Command (Cont.)**

**Example**

Print the disk usage values for your job.

```
.DSK   
RD,WT=5,10  
RD,WT=474,12
```

Print the total disk usage values for job 50.

```
.DSK 50   
RD,WT=474,12
```

## DUMP Command

### Function

The DUMP command analyzes a file and provides printable output.

The DUMP commands are discussed in the DUMP specification in the TOPS-10 Notebook Set.

### Formats

DUMP/command/command/command...

DUMP @dev:file.ext [directory]

DUMP

Commands that appear in the DUMP command string are passed to the DUMP program. A DUMP command using a command file can also specify these commands. A DUMP command without arguments prints a short dump of your core area in the command file QUIKDM.CCL, that resides on device SYS:. The file produced is stored in SYS: and has a file name nnnDAE.TMP, where nnn is the job number with leading zeros.

### Characteristics

The DUMP command:

Requires LOGIN.

Leaves your terminal at monitor level.

Destroys your core image.

### Example

This command string writes a core image file named 030DAE.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; 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.

```
.DUMP/OUT:/TTY:/MODE:ASCII.SIXBIT/WIDTH:7.10/JUST:L.R-  
/RIGHTMA:26/D [3000 & 3004]RET
```

**E(xamine) Command****Function**

The E(xamine) command examines a core location in your area (high or low segment). This function is useful when debugging programs.

DAEMON will be called on execution of the E command if the page in question is paged out.

**Format**

E addr

Where: **addr** is required the first time the E command is used. The contents of the location are typed out in half-word octal mode. If **addr** is specified, the contents of the address are printed.

If **addr** is not specified, the contents of the location specified in a previous E or D command are printed.

**Characteristics**

The E command:

Requires LOGIN.

Leaves your terminal at monitor level.

Requires core.

**Example**

Deposit in location 141.

```
.D 266000 2616 141RET
```

Examine location 140.

```
.E 140RET
000140/ 265720 40011
```

Because the address is omitted, the deposit made is in the location of the last E command (that is, location 140).

```
.D 47000 1RET
```

Examine the location specified in the previous D command.

```
.ERET
000140/ 047000 000001
.
```

## SYSTEM COMMANDS AND PROGRAMS

### **EDIT Command**

This command and the program LINED are obsolete and unsupported.

**ENABLE Command****Function**

If you are not a privileged user, this command has no effect. However, if you are a privileged user (that is, you have the privilege word set in ACCT.SYS) you can turn your privileges off with the DISABLE command, and you can turn your privileges on with the ENABLE command. Your privileges are enabled automatically when you log in. Privileges allow you to use privileged monitor calls such as POKE and SPY.

**Format**

ENABLE

**Characteristics**

The ENABLE command:

Does not destroy your core image.

Leaves your terminal at monitor level.

Does not affect privileges associated with [1,2] or with being on the operator's terminal (OPR:).

**Example**

Run FILDDT. Look at memory area. Examine address 41. Privileges are required to examine this location.

```
.R FILDDT (RET)
FILE: /M (RET)
41/ JSR3600
^C
```

Disable your privileges:

```
.DISABLE (RET)
```

Run FILDDT. Examine 41. Your privileges are disabled; therefore, 0 is printed rather than the contents of 41.

```
.R FILDDT (RET)
FILE: /M (RET)
41/ 0
^C
```

Enable your privileges:

```
.ENABLE (RET)
```

Again, examine memory. The contents of 41 are displayed.

```
.R FILDDT (RET)
FILE: /M (RET)
41/ JSR3600
^C
.
```

## EOF Command

### Function

The EOF command writes an end-of-file mark on the specified magnetic tape. This command runs the COMPIL program, which interprets the command before running the PIP program.

### Format

EOF MTXnnu:

Where: **MTXnnu:** specifies the magnetic-tape unit.

More than one magnetic-tape unit can be specified in the command string by separating the specifications with commas. Refer to Section 1.8.1 for a description of device name formats.

### Characteristics

The EOF command:

Requires LOGIN.

Leaves your terminal at monitor level.

Destroys your core image.

### Example

Write end-of-file mark on magnetic tape MTA2:.

.EOF MTA2: **RET**

## EXECUTE Command

### Function

The EXECUTE command compiles the specified source files, if necessary, loads the generated REL files into core, and begins execution of the program. The system determines the proper language compiler to use from the source file extensions or from switches you specify 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, the system does not compile the file unless you request this explicitly with a switch.

This command is equal to issuing the LOAD and START commands.

Each time you issue a COMPILE, LOAD, EXECUTE, or DEBUG command, the system remembers the command with its arguments in a temporary file on disk or in core. Therefore, when you issue one of these commands without specifying any arguments, the system uses the arguments that it saved in the temporary file. (Refer to Appendix C.)

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

This command runs the COMPIL program, which interprets the command before running the appropriate language compiler for LINK.

### Format

EXECUTE file-spec

Where: file-spec is a single file specification or a string of file specifications separated by commas. A file specification consists of a device name, a file name with or without an extension, and a directory name. (Refer to Section 1.8.)

You can use the following switches to modify the command string. These switches can be temporary or permanent switches unless otherwise stated. (Refer to Section 1.7.4.)

Switch	Function
/ALGOL	Compiles the file with ALGOL. Assumed for files with the extension of .ALG.
/BINARY	Generates a binary file for each file compiled. The file name for the binary file follows the standard conventions for determining the name of the output file. (Refer to the COMPILE command.) The file name extension is .REL. This is the default action.
/BLISS	Compiles the file with BLIS10. Assumed for files with the extension of .Bl0 or .BLI. This switch is not supported.

## SYSTEM COMMANDS AND PROGRAMS

### EXECUTE Command (Cont.)

Switch	Function
/C68 /C74	Runs the appropriate COBOL compiler.
/COBOL	Compiles the file with COBOL. Assumed for files with the extension of .CBL.
/COMPILE	Forces a compilation of this file even if a binary file exists with a newer date and time than the source file. You can use this switch to obtain an extra compilation (for example, to obtain a listing) because the system does not perform compilation if the binary file is newer than the source file.
/CREF	Produces a cross-referenced listing file on the disk for each file compiled, for later processing by the CREF program. The file extension of the output file is .CRF. You can then list the files using the CREF command. However, with COBOL files, the system appends the cross-referenced listing to the listing file.
/DDT	Loads the program debugger DDT with the program.
/DEBUG:(arg, arg,...)	Passes the arguments to FORTRAN. Refer to the <u>FORTRAN Programmer's Reference Manual</u> .
/DLIST	Produces a .LST file in your directory area. You can output the file to the line printer with the PRINT command.
/F40 /F10	Obsolete
/FOROTS /FORSE	Obsolete
/FORTRAN	Compiles the file with a FORTRAN compiler. Assumed for files with the extension of .F4 and .FOR and all files with nonrecognizable compiler extensions if FORTRAN is the standard compiler. You need to specify this switch if the file has a nonrecognizable compiler extension and FORTRAN is not the standard compiler or is not the current default. (For example, EXE/ALGOL FIL1, FIL2, FIL3/FORTRAN.)



## SYSTEM COMMANDS AND PROGRAMS

### EXECUTE Command (Cont.)

Switch	Function
/FUDGE: file-spec	<p>Creates a disk file that contains the names of the .REL files produced by the command string. When you issue the FUDGE command, PIP reads this file to generate a library .REL file. (Refer to the FUDGE command description.) The argument to this switch is:</p> <p>dev:file.ext[proj,prog]</p> <p>dev: is the device on which to write the file. If you omit the device, the system assumes DSK.</p> <p>file.ext is the name of the library file. The file name is required. If you omit the extension, the system assumes .REL.</p> <p>[proj,prog] is the directory in which to place the file. Your directory is assumed if you do not specify a directory.</p> <p>This switch is permanent in the sense that it pertains to all .REL files generated by the command string.</p>
/K?10	<p>Designates the processor where the program will execute once it has been loaded. The variable (?) can be A, L, or I. These switches are necessary for FORTRAN-10 programs because the compiler generates different code for the processors. The default is the processor on the computer executing the command.</p>
/LIBRARY	<p>Loads 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. (See the <u>LINK Programmer's Reference Manual</u>.)</p>
/LINK	<p>Obsolete</p>
/LIST	<p>Generates a listing file for each file compiled. The extension of the output file is .LST. The system automatically spools it when you log out. The complement of this switch is /NOLIST.</p>
/LMAP	<p>Produces a loader map during the loading process containing the local symbols.</p>
/LOADER	<p>Obsolete</p>
/MACRO	<p>Assembles the file with MACRO. Assumed for files with extensions of .MAC.</p>

## SYSTEM COMMANDS AND PROGRAMS

### EXECUTE Command (Cont.)

Switch	Function
/MACY11	Assembles the file with MACY11. Assumed for files with extensions of .Pl1. This switch is not supported.
/MANTIS	Compiles the file with MANTIS debugging information. This switch affects Fortran-40 files only. This switch is not supported.
/MAP	Produces 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 your disk area with either the file name specified by you (for example, /MAP:file) or the default file name nnnLNK.MAP. 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	Runs the appropriate language compiler from the experimental system library (device NEW:) area [1,5]. If the compiler does not exist on device NEW:, COMPIL tries to obtain it from device SYS:.. (Refer to the Restriction.)
/NOBINARY	Does not generate binary files. Unless you issue this switch, the system generates binary files. This switch, when combined with the /CREF or /LIST switch, is useful when you compile programs solely for the purpose of generating listings.
/NOCOMPILE	Does not force a compilation on a source file whose date is not as recent as the date on the binary file. This switch is not the same as the /REL switch, which turns off all compilations, even if the source file is newer than the .REL file. Complement to the /COMPILE switch. /NOCOMPILE is the default action.
/NODEBUG	Does not pass previously specified arguments to FORTRAN.
/NOLIST	Does not generate listing files. This is the default action.
/NOMANTIS	Compiles the file without MANTIS debugging information. This switch affects Fortran-40 programs only. This switch is not supported.
/NOOPTIMIZE	Does not optimize the object code of FORTRAN programs.
/NOSEARCH	Loads all routines of the file whether the routines are referenced or not. Because this is the default action, this switch is used to turn off library search mode (/LIBRARY).

## SYSTEM COMMANDS AND PROGRAMS

### EXECUTE Command (Cont.)

Switch	Function
/OLD	Runs the appropriate language compiler from the system library of old programs (device OLD:) which resides on the disk area [1,3]. If the compiler does not exist on device OLD:, COMPIL tries to obtain it from device SYS:. (Refer to the Restriction.)
/OPTIMIZE	Optimizes the object code of FORTRAN source programs.
/REL	Uses the existing .REL files although newer source files might be present.
/SAVE	Saves the core image of the loaded program.
/SEARCH	Loads the files in library search mode. The action is identical to that of the /LIBRARY switch.
/SELF	Runs the appropriate language compiler from device DSK: instead of from the system library (device SYS:). This switch is useful if you keep a private copy of a compiler in your own disk area. (Refer to the Restriction.)
/SNOBOL	Compiles the file with SNOBOL. Assumed for files with an extension of .SNO. This switch is not supported.
/SSAVE	Saves the core image of the loaded program in a sharable executable file.
/SYS	Runs the appropriate language compiler from the system library (device SYS:) area [1,4]. This is the default action.

#### Restriction

You cannot specify compilers from different structures in the same command string. For example, the following is invalid:

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

However, the following is valid:

```
.COMPILE PARTA.FOR/NEW  
.COMPILE PARTB.FOR/OLD  
.EXECUTE/REL PARTA,PARTB
```

**EXECUTE Command (Cont.)****Characteristics**

The EXECUTE command:

Requires LOGIN.

Places your terminal at user level.

Runs the appropriate compiler or assembler and linking loader, destroying your original core image.

Starts the execution of the compiled and loaded program.

**Example**

Print a test program.

```
.TY PROG.FOR(RET)
      TYPE 10
      10 FORMAT (' TESTING EXECUTION')
      END
```

Execute the program.

```
.EXECUTE PROG.FOR(RET)
FORTRAN:  PROG
MAIN
LINK:  LOADING
[LNKXCT PROG EXECUTION]

TESTING EXECUTION

END OF EXECUTION
CPU TIME:  0.02 ELAPSED TIME:  0.05
EXIT
.
```

Message shows the time parameters of your job.

## FILCOM Program

### Function

The FILCOM program compares two files and writes 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.

### Format

```
R FILCOM
*output file-spec=input-file-spec,input-file-spec/switch
```

Where: \* is the system prompt.

file-spec is a file specification in the standard format, namely:

```
dev:file.ext[directory]
```

output-file-spec specifies the name and area on which the results should be stored.

input-file-spec is a list of the files to be compared.

switches can follow the input file-specs. The switches are listed below.

### Defaults

1. If you omit the output specification, the output device is assumed to be TTY:. If you omit the output file specification, you must still type the equal sign.
2. If you specify only the output file name, the default output device is DSK:.
3. If you omit the output file name, the second input file name is used, unless it is null. In this case, the file name FILCOM is used.
4. If you omit the output file extension, .SCM is used on a source compare and .BCM is used on a binary compare.
5. If you omit the [directory] in either the input or output file specifications, your default directory is assumed.
6. If you omit an input device name, it is assumed to be DSK:.
7. If you omit the file name of the second input file, it is taken from the first input file.

**FILCOM Program (Cont.)**

8. A dot following the file name of the second input file 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. (that is, with null extension), use the following command string:

```
R FILCOM(RET)
*=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 you a method of listing a binary file. (Refer to Example 4.)

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

Switch	Binary Compare
/H	Prints a list of switches available (help text from device HLP:).
/nL	Specifies 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, prints the message ?FILES ARE DIFFERENT, but does not list the differences. This switch is useful when batch control files test for differences but do not want a log file of these differences.
/nU	Specifies 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	Compares 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	Expands .SAV files before comparing them in binary mode. This action removes differences resulting from zero compression.

## SYSTEM COMMANDS AND PROGRAMS

### FILCOM Program (Cont.)

Switches	Source Compare
/A	Compares files in ASCII mode. This switch is used to force comparison of two files to be done in ASCII.
/B	Compares blank lines. Without this switch, blank lines are ignored.
/C	Ignores 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	Prints a list of switches available (help text from device HLP:).
/nL	Specifies 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 to be identical. When a match is found, all differences occurring before the match and after the previous match are listed. In addition, the first line of the current match is listed 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	Prints the message ?FILES ARE DIFFERENT, when the files are different, but does not list the differences.
/S	Ignores spaces and tabs.
/U	Compares in update mode. This means that the output file consists of the second input file with vertical bars (or backslashes 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 following extensions.

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

### FILCOM Program (Cont.)

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

1. Files with extensions .SHR and .HGH are assumed to be high-segment files. Because 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.
3. Files with the extension .EXE are assumed to be in .EXE format.

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, file name, extension, and creation date of each input file are listed before the differences are listed. 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 (refer to the 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 listed 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 1st file-word 2nd file-word XOR of both words

If the exclusive OR (XOR) of the two words differs only in the right half, the third word listed 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 listed.



## FILCOM Program (Cont.)

### Characteristics

The R FILCOM command:

Places your terminal at user level.

Destroys your core image.

Requires LOGIN.

### Example

You have the following two ASCII files on disk:

```
.TYPE FILE1.TST (RET)
```

```
A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z
```

**FILCOM Program (Cont.)**

```
.TYPE FILE2.TST   
A  
B  
C  
G  
H  
I  
1  
2  
3  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
4  
5  
W  
X  
Y  
Z
```

To compare the two files and print the differences on your terminal, use the following sequence.

Run the FILCOM program.

```
.R FILCOM 
```

SYSTEM COMMANDS AND PROGRAMS

**FILCOM Program (Cont.)**

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

```
*=FILE1.TST,FILE2.TST(RET)
```

```
FILE 1) DSKC:FILE1.TST[27,5434] CREATED: 0825 10-JAN-80  
FILE 2) DSKC:FILE2.TST[27,5434] CREATED: 0826 10-JAN-80
```

```
1)1 D  
1 E  
1) F  
1) G  
****  
2)1 G  
*****  
1)1 J  
1) K  
1) L  
1) M  
1) N  
****  
2)1 1  
2) 2  
2) 3  
2) N  
*****  
1)1 W  
****  
2)1 4  
2) 5  
2) W  
*****
```

```
%FILES ARE DIFFERENT
```

SYSTEM COMMANDS AND PROGRAMS

**FILCOM Program (Cont.)**

To compare the two files and print the differences on the line printer, use the following command. In this example the number of successive lines that determines a match has been set to four with the /4L switch.

```
*/4L = FILE1.TST,FILE2.TST RET  
FILE 1) DSKC:FILE1.TST[27,5434] CREATED 0825 10-JAN-1980  
FILE 2) DSKC:FILE2.TST[27,5434] CREATED 0826 10-JAN-1980
```

```
1)1 D  
1) E  
1) F  
1) G  
1) H  
1) I  
1) J  
1) K  
1) L  
1) M  
1) N  
****  
2)1 G  
2) H  
2) I  
2) 1  
2) 2  
2) 3  
2) N  
*****  
1)1 W  
****  
2)1 4  
2) 5  
2) W  
*****
```

%FILES ARE DIFFERENT

**FILCOM Program (Cont.)**

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(RET)
*/U=FILE1.TST,FILE2.TST(RET)
  A
  B
  C
|  G
  H
  I
|  1
|  2
|  3
  N
  O
  P
  Q
  R
  S
  T
  U
  V
|  4
|  5
  W
  X
  Y
  Z

%FILES ARE DIFFERENT
```

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

## FILE Command

### Function

The FILE command remotely controls DECTape-to-disk and disk-to-DECTape transfers on operator-handled DECTapes.

There are seven functions that can be performed by the FILE command.

Format	Function
FILE C	Check
FILE D	Delete
FILE F	File
FILE L	Read
FILE R	Retrieve
FILE W	Wait
FILE Z	Zero

The C and W functions are the only requests that are performed immediately. Your terminal and job are free to proceed before the request is completed, except for batch jobs, which cannot continue until execution is complete.

### Formats

1. FILE C

Checks the queue of FILE requests to determine if any of your requests are still pending. There is no argument to the command in this format. Pending requests for your job will be printed on your terminal.

2. FILE D,tape-id,file.ext,file.ext,...

Deletes the specified files from DECTape. This command requires tape identification and a list of file names as arguments. The tape-id is a 1- to 6-character alphanumeric name that identifies the tape. After the files are deleted, an automatic FILE L is performed.

3. FILE F,tape-id file-spec,file-spec,...

Copies the specified files onto the specified DECTape. This command requires a tape identification and list of file specifications as arguments. The file specifications can include an explicit file structure name and project-programmer number so that you can copy files from a disk area other than your own. You do not have to specify the device and project-programmer number of subsequent file specifications if they do not change. That is, you must specify the programmer number (for example, [,104]) if the file to be copied has the same project number as yours and you must specify the project number ([41,]) if you are copying files from another project.

**FILE Command (Cont.)**

The protection of the disk file is checked to see if the file can be read. In addition, the protection of the DECTape directory file (tape-id.DIR) is checked to see if you can update it. If there is not enough room on the DECTape to copy an entire file, that portion (if any) that has been written so far is deleted and an error message is placed in the directory file. When the files have been successfully copied, an automatic FILE L is performed.

When the wildcard construction is used with the FILE F and FILE Z commands, your job's entire search list is used. That is, all files matching the construction are transferred from all structures in the search list.

In most cases you do not need to specify which file structures the files are on because UMount determines this (with LOOKUPS) and passes the information to OMount. With the FILE F and FILE Z functions, if a file structure is not specified and the specified file exists on more than one structure, the first one in your search list is copied.

4. FILE L,tape-id

Reads the directory of a DECTape and writes it into your disk area as an ASCII file with the file name tape-id.DIR. The file is placed on the first file structure that can be written on in your search list, as long as that structure has an area that matches your project-programmer number. Tape-id is a 1- to 6-character alphanumeric name that is used to identify the tape. You can then print the directory with the DIRECTORY command. The format of the DECTape directory is similar to the directory file written by the DIRECTORY command. If errors occur while the FILE command is being processed, the system writes error messages into the directory file tape-id.DIR.

5. File R,tape-id,dev:file.ext,dev:file.ext...

Transfers the specified files from your DECTape to the disk. This command requires tape-id and a list of file names as arguments. If the specified files already exist in the disk area, they are superseded, if their protection code allows it. If the specified files do not exist, they are created on the first file structure in your job's search list for which creation is permitted. After files are transferred, an automatic FILE L is performed.

If you have a search list containing multiple file structures, the asterisk construction (when used with the FILE R command) can cause files to be created rather than superseded. For the FILE R function, when no file structure is specified, the files are copied onto the first file structure in your search list on which you are 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.

## FILE Command (Cont.)

### 6. FILE W

Waits until all of your pending requests are processed before allowing your job to continue. If there are pending requests, the message "WAITING... TYPE 2 ^C'S TO EXIT" is printed on your terminal. Control returns to your job when all requests have been processed. You can type two CTRL/Cs if you decide not to wait.

### 7. FILE Z,tape-id,dev:file.ext[proj,prog],dev:file.ext[proj,prog]...

Deletes the files in the directory of the DECTape before the files are copied and then performs the same operations as the FILE F command. This command requires tape-id and can have a list of file specification arguments. After the files are copied, an automatic FILE L is performed.

When you use the wildcard construction with the FILE F and FILE Z commands, your job's entire search list is used. That is, all files matching the construction are transferred from all structures in the search list.

The wildcard constructions can be used, but generic DSK: can define many file structures; the specific file structure is chosen as follows:

When you use the wildcard construction for the file name or extension, the first structure in your search list that you can access is used.

If you do not use the wildcard construction and the file exists, the first file structure in the search list that contains the specified file is used, unless overridden by a default. (Refer to the examples.) If the file does not exist, the standard structure is used.

### Examples

In the following examples, your search list is as follows:

```
SORT: ,DSKA: /NOCREATE,DSKB: ,DSKC:
```

You are user 10,3421, with UFDs on DSKA:, DSKB:, and DSKC:, and the file EX.1 exists on each of these three structures. User 10,4072 has the file EX.2 in his area on DSKB: and on DSKC:.

1. This command requests that the operator mount DECTape 1, that the file EX.1 in your area be copied onto it, that the file EX.2[10,4072] also be copied onto the DECTape, that a directory of the DECTape be written in your area, and that the operator dismount the DECTape.

The directory is written on the first structure encountered in the search list that is both writeable and on which you have a UFD. In the preceding example, you do not have a UFD on the first structure in your search list (SORT:), you cannot create new files on the second structure in your search list (DSKA:), and so the directory is written on the third structure in your search list (DSKB:).



## SYSTEM COMMANDS AND PROGRAMS

### FILE Command (Cont.)

The file EX.1 on DSKA: will be copied only onto the DECTape because the copy on DSKA: was encountered first. The file DSKB:EX.2[10,4072] will be copied onto the DECTape because the copy on DSKB: was encountered first.

```
.FILE F,TAPE1,EX.1,EX.2[10,4072] (RET)
REQUEST QUEUED
1. F JOB30 TTY11 10,3421 TAPE1
DSKB:,DSKA0:EX.1,DSKB0:EX.2[10,4072]
1 COMMAND IN QUEUE
.
```

2. In this example, you specified that DSKC: be copied from both areas. DSKC: was typed only once because a device given in a file specification remains in effect for subsequent file specifications in the same command, unless another device is specified. When you omit the project number the default is your project-programmer number. Also the project-programmer number can be specified either before the file name (as in this example) or after the file name (as in the preceding example).

```
.FILE F,TAPE2,DSKC:EX.1,[,4072]EX.2 (RET)
REQUEST QUEUED
2. F JOB30 TTY11 10,3421 TAPE2
DSKB:DSKC0:EX.1,DSKC0:EX.2[10,4072]
2 COMMANDS IN QUEUE
.
```

The response from the FILE C command indicates to you which of your requests have not yet been processed. In this case, both of your requests are still pending.

```
.FILE C (RET)
1. F JOB30 TTY11 10,3421 TAPE1
DSKB:,DSKA0:EX.1,DSKB0:EX.2[10,4071]
2. F JOB30 TTY11 10,3421 TAPE2
DSKB:,DSKC0:EX.1,DSKC0:EX.2[10,4072]
2 COMMANDS IN QUEUE
.
```

The FILE R command uses the same algorithm as the FILE F and Z command for determining the device on which to write the directory. The file EX.1 is written on the first file structure in your search list. The file EX.1 already exists on DSKA:, DSKB:, and DSKC:. DSKA: is NOCREATE, so the file EX.1 is written onto DSKB, superseding the EX.1 already on DSKB:.

```
.FILE R,TAPE1,EX.1 (RET)
REQUEST QUEUED
1. R JOB24 TTY11 10,3421 TAPE1
DSKB:,DSKB:EX.1
1 COMMAND IN QUEUE
.
```

SYSTEM COMMANDS AND PROGRAMS

FILE Command (Cont.)

3. When the wildcard construction is used, UMount uses the entire search list to determine what files to copy for FILE F and FILE Z commands, whether or not you have a UFD on a particular structure. UMount passes the construction, along with each structure in the search list, to OMount.

```
.FILE F,TAPE3,E?.* (RET)
REQUEST QUEUED
2. F JOB24 TTY11 10,3421 TAPE3
DSKB:,SORT:E?.*,DSKA:E?.*,DSKB:E?.*,DSKC:E?.*
DSKC:E?.*
2 COMMANDS IN QUEUE
.
```

## FINISH Command

### Function

The FINISH command terminates any input or output currently in progress on the specified device, closes any open files, and deassigns the device. This command completely disassociates a device from your job and prevents you from continuing the program. If you want to continue your program after ending I/O, use the DEASSIGN command.

### Format

FINISH dev:

Where: **dev:** is 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 your job's controlling terminal, and any logical name associated with the controlling terminal is cleared.

Refer to Section 1.8.1 for a description of device names.

### Characteristics

The FINISH command:

Requires LOGIN.

Leaves your terminal at monitor level.

Requires core.

### Restrictions

When running a program, you cannot continue your program after a FINISH if the device was initialized, but you can start the program from the beginning or enter DDT.

### Example

Assign a card reader to your job.

```
.ASSIGN CDR:RET
```

Card reader number 1 is assigned.

```
CDR261 ASSIGNED
```

**FINISH Command (Cont.)**

Show the devices that are busy.

.SYSTAT B

BUSY DEVICES:			
DEVICE	JOB	WHY	LOGICAL
TTY144	18	INIT	
TTY235	18	INIT	
LPT260	18	INIT	
LPT261	18	INIT	
LPT263	18	INIT	
CDR260	18	INIT	
CDR261	29	AS	

Card reader number 1 is busy because it is assigned to your job.

Finish using the card reader.

.FINISH CDR

Again, show the busy devices.

.SYSTAT B

BUSY DEVICES:			
DEVICE	JOB	WHY	LOGICAL
TTY144	18	INIT	
TTY235	18	INIT	
LPT260	18	INIT	
LPT261	18	INIT	
LPT263	18	INIT	
CDR260	18	INIT	

Card reader 1 is not in the list of busy devices. Therefore, it is available to other users.

**FUDGE Command****Function**

The FUDGE command creates a library .REL file from a temporary file generated by a previous COMPILE, 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. (See the MAKLIB User's Guide and the LINK specification in the TOPS-10 Notebook Set for descriptions of library .REL files.)

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

**NOTE**

Because the COMPIL program sorts 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 you use the /FUDGE switch, you must issue the FUDGE command before you issue any other command that runs PIP (for example, TYPE and COPY). There is only one temporary file that contains information generated by the COMPILE-class commands that run PIP. If you do not use the FUDGE command, the information in the temporary file will be superseded by the information generated by any other command that runs PIP.

**Format**

FUDGE

**Characteristics**

The FUDGE command:

Leaves your terminal at monitor level.

Destroys your core image.

**Example**

Compile the files PROG,NUMBER, and NUMB2.

```
.COMPILE/FUDGE:LIBRAR PROG.FOR,NUMBER.FOR,NUMB2.FOR(RET)
```

Create a file named LIBRAR that contains the .REL files generated by the COMPILE command.

```
.FUDGE(RET)
```

### FUDGE Command (Cont.)

A DIRECTORY command shows the files created less than an hour ago, among them: LIBRAR.REL.

.DIR/SIN:-1<sup>RET</sup>

PROG	REL	1	<055>	4-DEC-79	DSKC:[27,5055]
NUMBER	REL	1	<055>	4-DEC-79	
NUMB2	REL	1	<055>	4-DEC-79	
LIBRAR	REL	3	<055>	4-DEC-79	

## GET Command

### Function

The GET command loads a core image from a retrievable storage device but does not execute it.

This command clears all of your core. However, programs should not count on this action and should explicitly clear those areas of core that are expected to contain zeros. (That is, programs should be self-initializing.) This action allows programs to be restarted by a CTRL/C START sequence without another GET command.

### Format

GET file-spec core

Where: **file-spec** is a file specification in the standard format.

**core** is the amount of core you want to reserve for the program. The core argument is a decimal number followed by an optional K or P for blocks or pages, respectively. If you do not specify either K or P, K is assumed. If you do not specify a core argument, the default is the minimum amount of core needed to load the program.

### Characteristics

The GET command:

Destroys your core image.

Leaves your terminal at monitor level.

Does not function during memory I/O.

### Example

Load a program.

```
.LOAD PROG.FOR RET
FORTRAN: PROG
MAIN.
LINK: LOADING
EXIT
```

Save the executable format.

```
.SAVE RET
PROG SAVED
```

Use GET to bring the program into memory.

```
.GET PROG RET
JOB SETUP
```

**GET Command (Cont.)**

START execution of the program.

.START (RET)

TESTING EXECUTION

END OF EXECUTION

CPU TIME: 0.02 ELAPSED TIME: 0.17

EXIT

.



## GLOB Program

### Function

The GLOB program reads multiple binary files and produces an alphabetical cross-referenced list of all the global symbols (symbols accessible to other programs) that it encounters. This program also searches files in library search mode, checking for globals, if the program file was loaded by LINK in library search mode. (See the LINK specification in the TOPS-10 Notebook Set.)

The GLOB program has two phases of operation:

1. Phase one scans the input files and builds an internal symbol table.
2. Phase two produces output based on the symbol table.

Because of these phases, you can enter 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.

### Formats

R GLOB

\*output file-spec=input file-spec,...(ESC)

Where: \* is the program prompt.

R GLOB

followed by one or more input commands in the form:

file-spec, file-spec,...(RET)

and then one or more output commands in the form:

output file-spec=(ESC)

When you separate your input to GLOB into input commands and output commands, the input commands contain only input specifications. The output commands contain 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.

**GLOB Program (Cont.)****Defaults**

If the device is omitted, it is assumed to be DSK:. However, if the entire output specification is omitted, the output device is TTY:.

If the output file name 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 file names are required.

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 file name.

If the directory area [directory] is omitted, your default directory is used.

An ESCape (or ALTmode) terminates the command input and signals GLOB to print the cross-referenced listing. A listing is not printed until GLOB encounters an ESCape. Press ESCape 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**

Job switches control the type of global listings to be printed. You precede each switch with a slash. If you issue several switches, they should 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 printed. The following is a list of the switches and their functions:

Switch	Function
/A	Prints all global symbols. This is the default if no switches are specified.
/E	Prints only erroneous (multiply defined or undefined) symbols.
/F	Prints nonrelocatable (fixed) symbols only.
/H	Prints a list of the switches available from HLP:GLOB.HLP.
/L	Scans programs only if they contain globals previously defined and not yet satisfied (library search mode).
/M	Turns off scanning mode resulting from the /L switch.
/N	Prints only symbols that are never referenced.

**GLOB Program (Cont.)**

Switch	Function
/P	Prints all routines that define a symbol to have the same value. The routine that defines the symbol first is printed, followed by a plus (+) sign. Subsequent routines that define the symbol are printed, preceded by a plus sign.
/Q	Suppresses the printing of subsequent defining routines, which result from the /P switch.
/R	Prints only relocatable symbols.
/S	Prints symbols with nonconflicting values that are defined in more than one program.
/X	Suppresses the header page when the output device is not your terminal, and includes the header when it is your terminal. Without this switch, the header is printed on all devices except your terminal. The listing header is in the following format:

F S OV D R

Where: F is a flag.

The flags are:

Flags	Meaning
M	Multiply defined symbol (all values are shown).
N	Never referred to (that is, was not declared external in any of the binary programs).
S	Multiply specified symbol (that is, defined in more than one program but with nonconflicting values). The name of the first program in which the symbol was encountered is followed by a plus sign.
U	Undefined symbol.
S is a symbol. Symbols listed are in alphabetical order according to their ASCII collating sequence. The octal value is followed by a apostrophe (') if the symbol is relocatable. The value is then relative to the beginning of the program in which the symbol is defined.	
OV is the octal value.	
D is where the symbol is defined.	
R is where the symbol is referenced.	

**GLOB Program (Cont.)****Characteristics**

The R GLOB command:

Requires LOGIN.

Places your terminal at user level.

Destroys your core image.

**Example**

Run the GLOB program.

All global symbols in the program MAIN (on DSK:), SUB40 (on DTA2:), and SUB50 are printed on the line printer. Each symbol is printed with its value, the program in which it is defined, all programs in which it is referenced, and any error flags.

```
.R GLOB 
```

```
*LPT:=MAIN.REL,DTA2:SUB40,SUB50 
```

The programs to be scanned are BATCH.REL and DATA.REL on DTA4, NUMBER.REL and CLASS.REL on DTA6:, and MATH.REL and LIBRAR. on DSK:.

```
*DTA4:BATCH.REL,DATA.REL,DTA6:NUMBER.REL,CLASS.REL 
```

```
*DSK:MATH.REL,LIBRAR.REL 
```

Print only nonrelocatable symbols on the line printer.

```
*LPT:=/F4 
```

Print only relocatable symbols in the file named SYMBOL in your default directory.

```
*DSK:SYMBOL=/R 
```

Print all erroneous symbols on the terminal. EXTSYM is an undefined symbol appearing in the program SUBRTE.

```
*TTY:=/E 
```

```
U EXTSYM SUBTRE
```

Return to monitor level.

```
*^C
```

```
.
```

## HALT Command

### Function

The HALT command (or CTRL/C) stops your job.

### Formats

HALT

(CTRL/C)

CTRL/C echoes on your terminal as ^C.

### Characteristics

The HALT or CTRL/C command:

Places your terminal at monitor level.

Does not require LOGIN.

### Examples

1. Request a list of your directory.

```
.DIR 
PATH1.RNO 3 <055> 2-MAR-79 DSKC:[27,5434]
SWITCH.INI 1 <055> 1-JUN-79
CALC.^C
^C
.
```

Type two CTRL/Cs to halt output.

2. Execute a program.

```
.EXECUTE PROG.FOR 
LINK: LOADING
[LNKXCT PROG EXECUTION]
^C
^C
.
```

Two CTRL/Cs return you to monitor level, interrupting program execution.

Use CCONTINUE to continue execution while remaining at monitor level.

```
.CCONTINUE 
```

Use HALT to interrupt execution. (See NOTE below.)

```
.HALT 
```

## HALT Command (Cont.)

Use CONTINUE to re-enter user level and continue program execution.

```
.CONTINUE (RET)
DONE
END OF EXECUTION
CPU TIME:6.12 ELAPSED TIME:45.73
EXIT
.
```

Program successfully completes execution.

### NOTE

It is necessary to halt execution of a program if your terminal is to enter or leave user level. Therefore, you must issue a HALT or CTRL/C at monitor level, after you have used the CCONTINUE or CSTART commands, for your terminal to enter user level.

## HELP Command

### Function

The HELP command prints information about system features on your terminal. You can select this information from the following file structures: HLP:, NEW:, OLD:, DOC:, and SYS:.

### Formats

#### HELP

Prints instructions for using the variations of the HELP command.

#### HELP \*

Lists the names of features for which HELP is available, all the monitor commands, all the monitor SET commands, the monitor SET TTY commands, the terminal types supported by TOPS-10, and the SET WATCH commands.

#### HELP name

Where: **name** is the name of a feature or program for which the monitor has a HELP file. Only the first six alphanumeric characters are read by the HELP program.

#### HELP dev:name

Where: **dev:** is the file structure that the file is on.  
**name** is the name of the program or feature for which you want help.

If the specified feature does not have a HELP file on the specified device, or if you do not specify the device, the devices are searched in the following order:

HLP:, SYS:, NEW:, OLD:, DOC:

You can change the search order by changing your job search list. Refer to the SETSRC command.

You can use the asterisk wildcard construction (\*) instead of the feature name, for information about all the HELP available on the specified device. (See examples.) You cannot use the asterisk (\*) to specify the device name.

## HELP Command (Cont.)

### Characteristics

The HELP command:

Destroys your core image.

Runs the HELP program.

Leaves your terminal at monitor level.

Does not require LOGIN.

### Examples

1. For information about the HELP command, type HELP.

```
.HELP RET
```

THE HELP COMMAND PRINTS INFORMATION ABOUT THE SYSTEM ON YOUR TERMINAL. USE ONE OF THE FOLLOWING CONSTRUCTIONS FOR SPECIFIC INFORMATION.

```
.HELP<RET>
```

PRINTS OUT THIS MESSAGE.

```
.HELP *<RET>
```

PRINTS OUT THE NAMES OF ALL SUBJECTS FOR WHICH THERE IS HELP. USE ONE OF THESE SUBJECTS AS THE NAME IN THE NEXT CONSTRUCTION. HELP \* ALSO PRINTS OUT THE NAMES OF THE SYSTEM COMMANDS, THE NAMES OF THE SET COMMANDS, THE NAMES OF THE SET DEFAULT COMMANDS, THE NAMES OF THE SET TTY COMMANDS, THE NAMES OF THE SET WATCH COMMANDS, AND THE SUPPORTED TERMINAL TYPES.

```
.HELP NAME<RET>
```

PRINTS ALL THE INFORMATION AVAILABLE ABOUT THE SUBJECT NAME. FOR EXAMPLE:

```
.HELP DIRECT<RET>
```

PRINTS INFORMATION ABOUT THE DIRECT COMMAND/PROGRAM.

```
.HELP LIB:*<RET>
```

PRINTS OUT THE NAMES OF ALL SUBJECTS FOR WHICH THERE IS HELP AVAILABLE FROM THE SPECIFIED LIBRARY DEVICE. THE STANDARD SYSTEM LIBRARY DEVICES ARE:

```
SYS:    NEW:    OLD:    HLP:    DOC:
```

FOR EXAMPLE:

```
.HELP NEW:*<RET>
```

PRINTS A LIST OF ALL THE INFORMATION AVAILABLE IN NEW:.

```
.HELP LIB:NAME<RET>
```

PRINTS INFORMATION ABOUT THE SPECIFIED SUBJECT IN THE SPECIFIED LIBRARY.



SYSTEM COMMANDS AND PROGRAMS

HELP Command (Cont.)

2. For a list of all the available HELP files and a list of monitor commands, type HELP \*.

.HELP \*<sup>(RE1)</sup>

HELP IS AVAILABLE FOR THE FOLLOWING:

ALQUIK	APL	APLSF	AVAIL	BACKUP	BASIC	BLIS10	BLIS32	BLKTRN
BLS36C	BOOT11	BOOTDX	BOOTGT	BOX	CAMCRP	CHANGE	COBDDT	COBOL
COBRG	COPYED	CPL	CREDIR	CREF	CRSCPY	CSORT	CSP10	DBMEND
DC72	DDT	DELFIL	DIR416	DIRECT	DISMOU	DSKRAT	DTADIR	DTELDR
ERSATZ	FAST	FGEN	FILCOM	FILDAE	FILE	FILEX	FIND	FNDBLD
FORDDT	FORDML	FORTRA	FRS	FUDGE2	GASP	GLOB	GRIPE	HELP
HERMES	HP35	HYPHEN	INI	INITIA	ISAM	JQ	KILL	KLAD10
KLBP	KSYS	LDVT20	LIBARY	LINED	LINK	LOGIN	LOGOUT	LPTSPD
MACRO	MACY11	MAIL	MAKLIB	MAKVFU	MATCH	MATHLA	MCSGEN	MEMO
MMIC	MODUNV	MOO	MOUNT.	MTCOPY	NETLDR	NETOPR	NETTST	NETWOR
NIK	NRUINP	NRUNOF	OLD	OMOUNT	OPR	OPSER	PASCAL	PATH
PCOPY	PIP	PLEASE	PMAKER	PPNLST	PRETTY	QDMP	QDUMP	QUEUE
QOOLST	REDUCE	RERUN	REV	RIPOFF	RSXFMT	RUNINP	RUNOFF	SAFEIO
SCDCP	SDLREM	SENGEN	SETSRC	SFCONV	SIMCOM	SIMDDT	SIMDIR	SIMRTS
SIMULA	SORT	SORTC	SOS	SOSX	SOUP	SPACE	SPELL	SPY
SUBNEW	SUBSYS	SY	SYSERR	SYSTAT	TAPE	TECO	TECOH	TRACK
TTT3D	TTYTST	TYPEIT	UFDREN		WDEL	WFLX	WHEEL	WLPT
WNDX	WPIP	XTEN	2102					

THE MONITOR HAS THE FOLLOWING COMMANDS:

ACCOUN	ALLOCA	ASSIGN	ATTACH	BACKSP	BACKTO	CANCEL	CCONTI	CLOSE
COJOB	COMPIL	CONTIN	COPY	CORE	CPUNCH	CREATE	CREF	CSTART
CTEST	D	DAYTIM	DCORE	DDT	DEALLO	DEASSI	DEBUG	DELETE
DETACH	DIRECT	DISABL	DISMOU	DO	DSK	DUMP	E	EDIT
ENABLE	EOF	ERROR	EXECUT	FILE	FINISH	FUDGE	GET	GOTO
HALT	HELP	IF	INITIA	JCONTI	KJOB	LABEL	LET	LIST
LOAD	LOCATE	LOGIN	MAKE	MERGE	MIC	MODIFY	MOUNT	NETWOR
NODE	NOERRO	NOOPER	NSAVE	NSSAVE	ON	OPERAT	OSAVE	OSSAVE
PATH	PJOB	PLEASE	PLOT	PRESER	PRINT	PROTEC	PUNCH	QUEUE
R	REASSI	REENTE	RENAME	RESOUR	REVIVE	REWIND	RUN	SAVE
SCHEDU	SEND	SESSIO	SET	SHOW	SILENC	SKIP	SOS	SSAVE
START	SUBMIT	SYSTAT	TECO	TERMIN	TIME	TPUNCH	TTY	TYPE
UNLOAD	USESTA	VERSIO	WHENEV	WHERE	XCHNGE	ZERO		

THE MONITOR HAS THE FOLLOWING SET COMMANDS:

BLOCKS	BREAK	CDR	CORMAX	CORMIN	CPU	CTEST	DATE	DAYTIM
DEFAULT	DEFER	DENSIT	DSKFUL	DSKPRI	FLOATI	HOSTES	HPQ	MEMORY
NODEFE	NOMESS	OPR	PHYSIC	RETRY	SCHEDU	SPOOL	TERMIN	TIME
TTY	VIRTUA	VMMAX	WATCH					

THE MONITOR HAS THE FOLLOWING SET TTY COMMANDS:

ALTMOD	BLANKS	COPY	CRLF	DEBREA	DEFER	DISPLA	ECHO	ELEMEN
FILL	FORM	GAG	HOLD	IGNORE	LC	NO	PAGE	REMOTE
RTCOMP	RTE	SLAVE	SPEED	TABS	TAPE	TIDY	TYPE	UC
WIDTH								

THE MONITOR SUPPORTS THE FOLLOWING TERMINAL TYPES:

DAS21	LA120	LA30	LA34	LA36	LA38	TTY	TTY33	TTY35
VT05	VT06	VT100	VT50	VT52	VT61	2741		

THE MONITOR HAS THE FOLLOWING SET WATCH COMMANDS:

DAY	FILES	MTA	READS	RUN	VERSIO	WAIT	WRITES	
-----	-------	-----	-------	-----	--------	------	--------	--

**INITIA Command****Function**

The INITIA command sets the parameters for your terminal. This command is issued automatically at system startup on certain designated terminals. You can reissue the command at any time. This command also starts certain system programs, when issued from particular terminals.

If you are logged in and you run INITIA, INITIA reads your SWITCH.INI file. Refer to Appendix B for a description of SWITCH.INI.

**Format**

INITIA arg arg ...

Where: arg can be one of the following arguments:

Argument	Function
BATCH	Prints the name of the batch processor that is running on the system (for example, GALAXY).
HELP	Prints a list of available keywords and their meanings.
KSYS	Tells you when timesharing will end.
NAME	Prints the system name.
NORUN	Suppresses the execution of any default program.
NOTICE	Prints the general operator notice.
SETTTY	Sets up your terminal characteristics.
STRUCTURES	Prints any operator notices about mounted structures.
TERMINET	Sets tab spacing identical to a TERMINET-300.
TEXT	Prints quick operator notice.
TTY	Prints the terminal characteristics for this terminal.

If you do not specify an argument, the default is NAME and STRUCTURES.

You can precede any of the above arguments with NO to suppress a default setting.

In addition to the above arguments, you can use the following arguments to INITIA in your SWITCH.INI file. They are effective for any job that is logged in. Each of the following arguments is equivalent to an argument to a SET TTY command. Many of the arguments can be preceded by NO to disable the defaults. These are included in the description of the function of the arguments.

SYSTEM COMMANDS AND PROGRAMS

**INITIA Command (Cont.)**

Each argument, equivalent SET TTY command, and function, is listed below.

Argument	Equivalent Command	Function
SETTTY	None	This argument has no equivalent command, but it must be present in the SWITCH.INI file for the other INITIA arguments to be effective.
ALTMODE	TTY ALTMODE	Converts ALTmode codes 175 and 176 to the ESCape character. NO ALTMODE restores the individual identities of codes 175 and 176.
BLANKS	TTY BLANK	Controls the output of blank lines. NO BLANKS suppresses the output of automatic, consecutive RETURNS at the end of output.
CRLF	TTY CRLF	Controls the automatic RETURN at the end of the terminal line. This is not effective at monitor level. NO CRLF suppresses the automatic RETURN.
DEBREAK	TTY DEBREAK	Notifies the system that the terminal has a feature that allows the system to lock the keyboard. Applies to Model 2741 terminals.
DEFER	TTY DEFER	Suppresses echoing to a video terminal until output is finished. NO DEFER allows characters typed to the system to be echoed during output to the terminal.
DISPLAY	TTY DISPLAY	Notifies the system that you have a display terminal. This characteristic can be used by programs that control output to the terminal.
ECHO	TTY ECHO	Controls echoing to the terminal. NO ECHO is used by local-copy terminals.
ELEMENT:xxx	TTY ELEMENT xxx	Changes the typing element number for Model 2741 terminals. This argument automatically sets NO LC.

SYSTEM COMMANDS AND PROGRAMS

**INITIA Command (Cont.)**

Argument	Equivalent Command	Function
FILL:x	TTY FILL x	Assigns filler class x to the terminal.
FORM	TTY FORM	Controls the output of line-feeds for formfeed and vertical tab characters. NO FORM instructs the system to output the line-feeds.
GAG	TTY GAG	Controls the reception of messages sent with the SEND command. NO GAG intercepts messages when your terminal is at user level.
LC	TTY LC	Informs the system that your terminal has lowercase ability. NO LC changes all lowercase characters to uppercase.
PAGE:xx	TTY PAGE xx	Sets the page length of your terminal. This controls the number of lines that will be output to your terminal before an automatic stop.
RCVSPEED:xxx	TTY SPEED xxx nnn	Changes the input speed of the terminal to be xxx. This argument effects the speed at which the monitor will receive characters from your terminal.
RTCOMP	TTY RTCOMP	Controls the use of CTRL/R and CTRL/T. Refer to Section 1.5 for the function of these control characters. RTCOMP turns off the control characters. NO RTCOMP turns them on.
SPEED:xxxx	TTY SPEED xxxx	Sets the speed of your terminal to xxxx baud. This construction sets the input and output speeds to be equivalent.
TABS	TTY TAB	Informs the system that your terminal has tabular ability. NO TABS instructs the system to simulate tab stops.

SYSTEM COMMANDS AND PROGRAMS

INITIA Command (Cont.)

Argument	Equivalent Command	Function
TAPE	TTY TAPE	Turns on the XON (CTRL/Q) and XOFF (CTRL/S) keys for paper tape reading. NO TAPE gives CTRL/Q and CTRL/S their normal function. TAPE is discussed in the <u>TOPS-10 Monitor Calls Manual</u> . CTRL/S and CTRL/Q are discussed in Section 1.5.
TYPE:xxxx	TTY TYPE xxxx	Informs the system that your terminal is of the type xxxx.
UC	TTY UC	Changes all the characters input from your terminal to uppercase.
WIDTH:xxx	TTY WIDTH xxx	Sets the width of the line on your terminal to xxx characters.
XMTSPEED:xxx	TTY SPEED nnn xxx	Changes the output speed of your terminal. This argument affects the speed at which the monitor sends characters to your terminal.

The arguments listed above are described more thoroughly in the description of the SET TTY command.

Your SWITCH.INI file can also contain any of the following arguments. These set a default system program to be run after the terminal characteristics are set. The arguments and the system programs they run are listed below.

Argument	Program
SYSDLT	SYSDLT
SYSDPA	SYSDPA
SYSDPB	SYSDPB
SYSDPY	SYSDPY
SYSHZL	SYSHZL
SYSVFT	SYSV52
SYSVSO	SYSV61

INITIA Command (Cont.)

Characteristics

The INITIA command:

Runs the INITIA program.

Does not require LOGIN.

Example

1. Show the characteristics of a generic (unspecified) terminal type:

```
.INITIA TTY (RET)
```

```
RZ024A KL#1026/1042 10:27:27 TTY31 SYSTEM 1026/1042
CONNECTED TO NODE KUL26 (26) LINE #31
JOB 49 USER MCCARRON,M [27,5055]
DSKC: KL1026 SYSTEM DISK DSKC
DSKB: KL1026 SYSTEM DISK DSKB
```

```
TYPE:TTY      NOLC      NOTABS      NOFORM
ECHO          CRLF      WIDTH:72    GAG
NODISPLA     FILL:1    NOTAPE      PAGE
BLANKS       NOLTMOD   RCVSPEED:2400 XMTSPEED:2400
NODEBREA     NOTIDY   NORTCOMP    NOSLAVE
NOHDX        REMOTE
```

Use the SET TTY TYPE command to specify your terminal type. Then use INITIA to show the terminal characteristics:

```
.SET TTY TYPE VT52 (RET)
```

```
.INITIA TTY (RET)
```

```
RZ024A KL#1026/1042 10:27:27 TTY31 SYSTEM 1026/1042
CONNECTED TO NODE KUL26 (26) LINE #31
JOB 49 USER MCCARRON,M [27,5055]
DSKC: KL1026 SYSTEM DISK DSKC
DSKB: KL1026 SYSTEM DISK DSKB
```

```
TYPE:VT52    LC      TABS      NOFORM
ECHO         CRLF      WIDTH:80  GAG
DISPLA      FILL:0    NOTAPE    PAGE:24
BLANKS      NOLTMOD   RCVSPEED:2400 XMTSPEED:2400
NODEBREA    NOTIDY   NORTCOMP    NOSLAVE
NOHDX       REMOTE
```

## JCONTINUE Command

### Function

The JCONTINUE command forces the specified job to continue if the job was in a CTRL/C state because of a device error.

### Format

JCONTINUE n

Where: n is the number of the job to be continued. This argument is required.

### Characteristics

The JCONTINUE command:

Places your terminal at monitor level.

Does not require LOGIN.

### Examples

1. Continue job 44:

```
.JCONTINUE 44 (RET)
```

.

2. Continue job 12:

```
.JCONT 12 (RET)
```

.

## KJOB Command

### Function

The KJOB command:

1. Stops all assigned I/O devices and returns them to the system's pool of available devices.
2. Returns all allocated memory to the system's pool of available memory.
3. Returns the job-number to the pool of available job-numbers.
4. Leaves your terminal at monitor level.
5. Prints run-time information for your job.
6. Completes all deferred spooling requests.

### Format

KJOB/switch

KJOB can be abbreviated to K. KJOB accepts one of the following four switches:

Switch	Function
/FAST	Logs out immediately and saves all files (including temporary files) as they are. This command is equal to an R LOGOUT command or a RUN monitor call to LOGOUT. Unpreserved temporary files (for example, nnn???.TMP where nnn is your job-number and the files reside in your disk area) are deleted. This is the default action of the KJOB command.
/BATCH	Deletes files only when you are over the logged-out quota, then deletes 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.
/HELP	Lists the KJOB switches and their meanings. This switch does not perform functions 1 through 4.
/NOMESSAGE	Causes all terminal output to be suppressed.



## SYSTEM COMMANDS AND PROGRAMS

### KJOB Command (Cont.)

#### Characteristics

The KJOB command:

Deassigns your terminal.

Stops all I/O devices that are assigned to your job.

Runs the LOGOUT program.

#### Example

```
.K/F RET  
JOB 99 USER CUSTER,L. [27,4072]  
LOGGED-OFF TTY64 AT 16:53:19 ON 20-JUN-79  
RUNTIME: 0:00:05, KCS:44, CONNECT TIME: 0:31:23  
DISK READS: 1534, WRITES: 88, BLOCKS SAVED: 2510
```

.

## LABEL Command

### Function

The LABEL command writes an identifier onto a DECTape. The identifier is stored on the tape itself and is printed when you print a directory of the tape with the DIRECT command. You should assign a unique label to each DECTape to avoid confusing one tape with another. You must use the ASSIGN command to access the tape before you can use LABEL.

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

### Format

LABEL dev: ^name^

Where: **dev:** is a physical or logical name that represents a DECTape.

^ is the delimiter of the DECTape identifier. The delimiter can also be quotes. If the identifier consists entirely of alphanumeric characters, the delimiters can be omitted.

**name** is a 1- to 6-character name to be used as the identifier. Any characters can be used except the delimiter.

### Restriction

You must use the ASSIGN command to access the tape before you can use LABEL.

### Characteristics

The LABEL command:

Leaves your terminal at monitor level.

Destroys your core image.

### Examples

1. Identify DECTape 4, which has the logical name TIME:

```
.LABEL TIME ^2:00^ (RET)
```

.

2. Identify DECTape 5 as NAME:

```
.LABEL DTA5: "NAME" (RET)
```

.

**LIST Command****Function**

The LIST command prints the contents of the named source file(s) on the line printer (LPT:). The output goes either to the line printer immediately or to the disk to be spooled to the line printer if the line printer is being spooled for this job. (Refer to the QUEUE and PRINT commands.) If the line printer is being spooled, the PRINT command is preferred over the LIST command because it saves time and disk accesses.

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

**Format**

LIST file-spec

Where: **file-spec** is a single file specification or a string of file specifications separated by commas. A file specification consists of a device name, a file name and extension, and a directory name. This argument is required. When a directory name precedes the file name, 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 to PIP.

**Characteristics**

The LIST command:

Leaves your terminal at monitor level.

Destroys your core image.

**Example**

Spool your line printer requests, print a file (FOR21.DAT) on the line printer, and look at the line printer queue.

```
.SET SPOOL LPT:(RET)
```

```
.LIST FOR21.DAT(RET)
```

```
.PRI(RET)
```

```
PRINTER QUEUE:
```

JOB NAME	REQ#	LIMIT	USER
*SYSTAT	61	12	MAROTTA[27,5434] ON UNIT:0

```
THERE IS 1 JOB IN THE QUEUE (1 IN PROGRESS)
```

```
.
```

## LOAD Command

### Function

The LOAD command translates the specified source files if necessary, runs the loader, and loads the generated .REL files into memory. The appropriate language compiler 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 you request it using the /COMPILE switch).

This command runs the COMPIL program, which interprets the command before running the appropriate language compiler and linking loader.

The LOAD command generates a core image but does not begin executing the program. After the LOAD command, you can START your program or SAVE the core image for future execution.

Each time the COMPILE, LOAD, EXECUTE, or DEBUG command is executed, the system remembers the arguments and switches. Therefore, if you issue one of these commands with no arguments, the system uses the stored arguments.

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

### Format

LOAD file-spec,file-spec,...

Where: **file-spec** is one or more file specifications separated by commas. A file specification consists of a device name, a file name with or without an extension, and a directory name. (Refer to Section 1.8.)

The following switches modify the command string.

Switch	Function
/ALGOL	Compiles the file with ALGOL. Assumed for files with the extension of .ALG.
/BINARY	Generates a binary file for each file compiled. The file extension of the output file is .REL. This is the default action.
/BLISS	Compiles the file with BLIS10. Assumed for files with the extension of .Bl0 and .BLI.
/C68 /C74	Runs the specified COBOL compiler.
/COBOL	Compiles the file with COBOL. Assumed for files with the extension of .CBL.

## LOAD Command (Cont.)

Switch	Function
/COMPILE	Compiles the 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 (for example, to obtain a listing of the compilation) because compilation is not performed if the binary file is newer than the source file.
/CREF	Produces a cross-referenced listing file on the disk for each file compiled, for later processing by the CREF program. The file extension of the output file 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.
/DDT	Loads the program debugger DDT with the program.
/DEBUG: (arg, arg,...)	Passes the specified arguments to FORTRAN. Refer to the <u>FORTRAN Programmer's Reference Manual</u> .
/DLIST	Produces a .LST file and stores it in your directory. You can obtain a listing of the file with the PRINT command.
/F10 /F40	Obsolete
/FORDDT	Loads the FORTRAN debugger with the program.
/FOROTS /FORSE	Obsolete
/FORTRAN	Compiles the file with a FORTRAN compiler. Assumed for files with the extension of .F4 and .FOR and all files with nonstandard compiler extensions. This switch is necessary if the file has a nonstandard compiler extension and FORTRAN is not the standard compiler or is not the current default.
/FUDGE: file-spec	Creates a disk file containing the names of the .REL files produced by the command string. When you give the FUDGE command, PIP reads this file to generate a library REL file. (Refer to the FUDGE command description.) The argument to the switch is:  dev:file.ext[directory]  dev: - the device on which to write the file. If you omit the device name, DSK: is assumed.

## SYSTEM COMMANDS AND PROGRAMS

### LOAD Command (Cont.)

- /FUDGE: (Cont.)
- `file.ext` - the name of the library file. The file name is required. If you omit the extension, it is assumed to be .REL.
- `[directory]` - the directory in which to place the file. Your default directory number is assumed if none is given.
- This switch is permanent in that it pertains to all .REL files generated by the command string.
- /K?10 Designates the machine on which the program will execute once it has been loaded. The ? can be A, I, or L.
- /LIBRARY Loads the files in library search mode. This mode causes a program in a special library 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 LINK documentation.)
- /LINK Obsolete
- /LIST Generates a disk listing file for each file compiled. The file extension of the output file is .LST. These files can be listed later with the PRINT command. If the line printer is being spooled for this job, the files are automatically printed. If you do not specify this switch, listing files are not generated.
- /LMAP Produces a loader map during the loading process that contains the local symbols.
- /LOADER Obsolete
- /MACRO Assembles the file with MACRO. Assumed for files with extension of .MAC.
- /MACY11 Assembles the file with MACY11. Assumed for files with an extension of .P11. This switch is not supported.
- /MANTIS Compiles the file with MANTIS. This switch affects Fortran-40 FORTRAN programs only. This switch is not supported.
- /MAP Produces a loader map during loading. After a library search of the default libraries, the map is written in your disk area with either the file name you specify (for example, /MAP:file) or with the default file name MAP.MAP. This switch is an exception to the permanent switch rule, because although it may be used as a permanent switch, it causes only one map to be produced.

## SYSTEM COMMANDS AND PROGRAMS

### LOAD Command (Cont.)

Switch	Function
/NEW	Runs the appropriate language compiler from the experimental system library (device NEW:) area [1,5]. If the compiler does not exist on device NEW:, COMPIL tries to obtain it from device SYS:. (Refer to Restriction.)
/NOBINARY	Does not generate binary files. This switch, when combined with the /CREF or /LIST switch, is useful when compiling programs only to generate listings.
/NOCOMPILE	Does not force a compilation of 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. Complement to the /COMPILE switch.
/NODEBUG	Does not pass previously specified arguments to FORTRAN.
/NOLIST	Does not generate listing files. This is the default action.
/NOMANTIS	Compiles the file without the MANTIS debugging information. This switch affects Fortran-40 programs only. This switch is not supported.
/NOOPTIMIZE	Does not optimize the object source code of a FORTRAN program.
/NOSEARCH	Loads all routines of the file whether the routines are referenced or not. Because this is the default, this switch turns off library search mode (/LIBRARY).
/OLD	Runs the appropriate language compiler from the system library of old programs (device OLD:) that resides on the disk area [1,3]. If the compiler does not exist on device OLD:, COMPIL tries to obtain it from device SYS:. (Refer to Restriction.)
/OPTIMIZE	Optimizes the object source code of a FORTRAN program.
/REL	Uses the existing .REL files although a newer source file might be present.
/SAVE	Saves the core image of the loaded program.
/SEARCH	Loads the files in library search mode. This switch is identical to the /LIBRARY switch.

**LOAD Command (Cont.)**

Switch	Function
/SELF	Runs the appropriate language compiler from device DSK: instead of from the system library (device SYS:). This switch is useful if you keep a private copy of a compiler in your disk area to test new features. (Refer to Restriction.)
/SNOBOL	Compiles the file with SNOBOL. Assumed for files with an extension of .SNO. This switch is not supported.
/SSAVE	Saves the core image of the loaded program in a sharable executable file.
/SYS	Runs the appropriate language compiler from the system library (device SYS:). This is the default action.

**Restrictions**

Once a language compiler has been specified from a particular area (for example, /SELF), it cannot be called from a different area within the same command string. The following is illegal:

```
.LOAD ITEM.CBL/SYS,ITEM02.CBL/SELF
```

However, the following is valid:

```
.COMPILE ITEM.CBL/SYS
.COMPILE ITEM02.CBL/SELF
.LOAD/REL ITEM,ITEM02
```

**Characteristics**

The LOAD command:

Leaves your terminal at monitor level.

Runs the appropriate compiler or language compiler and LINK, destroying your original core image.

**Example**

The following example shows how to LOAD and SAVE a program.

The DIRECTORY command shows all the files named TEST.

```
.DIR TEST.* (RET)
TEST FOR 1 <055> 10-SEP-79 DSKC: [27,5055]
```

LOAD the program. The monitor compiles and loads TEST.

```
.LOAD TEST.FOR (RET)
FORTRAN:TEST
MAIN.
LINK: LOADING

EXIT
```



SYSTEM COMMANDS AND PROGRAMS

**LOAD Command (Cont.)**

This command shows the old file and the new .REL file.

```
.DIR TEST.*RET  
TEST FOR 1 <055> 10-SEP-79 DSKC: [27,5055]  
TEST REL 1 <055> 5-DEC-79  
TOTAL OF 2 BLOCKS IN 2 FILES ON DSKC: [27,5055]
```

Save the loaded program.

```
.SAVE TESTRET  
TEST SAVED
```

A directory shows that the executable format has been saved in your directory.

```
.DIR TEST.*RET  
TEST FOR 1 <055> 10-SEP-79 DSKC: [27,5055]  
TEST REL 1 <055> 5-DEC-79  
TEST EXE 160 <055> 5-DEC-79 6(422)  
TOTAL OF 162 BLOCKS IN 3 FILES ON DSKC: [27,5055]  
.
```

**LOCATE Command****Function**

The LOCATE command logically establishes your job at a specified network node. When you log in, the logical location of your job is the same as the physical location. Therefore, this command is necessary only if you want to change your node (that is, your logical location). This command is used to change the default device list of the job, making devices at other nodes available to your job. For example, if you want to have the devices at node CHRIS perform all I/O for your job, you can issue the following command:

```
.LOCATE CHRIS
```

and the default I/O devices used will be those on node CHRIS.

The LOCATE command does not change the location of your job; the job runs on the node on which the job was started.

**Format**

```
LOCATE node-id
```

Where: **node-id** is the name or number of the node on which you want to be logically located.

An argument of 0 locates your job at the node where the job's command interpreter is. If you do not specify the node-id, the default is your physical node/station.

**Characteristics**

The LOCATE command:

- Leaves your terminal at monitor level.

- Is valid only in networks or in systems with a remote station.

- Does not destroy your core image.

**Associated Messages:**

If the LOCATE succeeds, the system prints a message in the form:

```
NODE node-name (node-num) sys-id date LOCATED
```

Where node-name is the name of the node, node-num is the number of the node, sys-id is the system identification, and date is the date the system was last reloaded.

If the node specified is not currently in communication with the network, the following message is printed:

```
%NODE(x) LOCATED BUT OFF-LINE
```

The job is successfully LOCATED, but the node is not presently accessible.

## LOCATE Command (Cont.)

### Examples

1. You logically locate the job at the node named LONDON, node number 3.

```
.LOCATE LONDON(RET)
NODE LONDON(3) RD020A KL10 SYS#1279 11-06-79 LOCATED
.
```

2. You attempt to locate to a node that the monitor does not recognize as part of the network. An error message is printed and your location is not changed.

```
.LOCATE TOKYO(RET)
?UNDEFINED NETWORK NODE
.
```

## LOGIN Command

### Function

The LOGIN command allows you to gain access to the system. The LOGIN command runs the LOGIN program. LOGIN accepts your project-programmer number and password, which you can obtain from your system administrator. LOGIN can accept a path specification.

To log in, type the LOGIN command, and press RETURN. LOGIN prompts you with a pound sign (#). Type your project-programmer number, and press RETURN. You can also type your project-programmer number immediately after typing LOGIN. After it receives your project-programmer number, LOGIN prints another prompt (PASSWORD:) on your terminal. Type your password, and press RETURN. You are finished logging in.

To specify a default path for disk I/O, type the path specification instead of your project-programmer number. A path specification more specifically identifies your directory area. It specifies either a user-file directory or a sub-file directory. Refer to Section 1.13 of this manual for more information about path specifications.

If you have detached your terminal from a job, and want to log in to another job, the LOGIN program asks if you want to attach to the old job or create a new job. For information about detaching and attaching jobs, see the ATTACH and DETACH commands. Example 2 shows the procedure for logging in to a new job after detaching the first.

### Format

LOGIN proj,prog/switches

Where: **proj,prog** stands for your project-programmer number.

You can type a path specification in place of the project-programmer number. This allows you to establish a default directory path for the job. The path specification is either a user-file directory or a sub-file directory. Directory paths are enclosed in brackets (for example: [27,5434,SFD]).

The following switches modify the command string. These switches can also be in your SWITCH.INI file. SWITCH.INI files are discussed in Appendix B.

Most parameters set by these switches can be changed by monitor commands after you log in. (Refer to the SET commands and the SETSRC program.)

## SYSTEM COMMANDS AND PROGRAMS

### LOGIN Command (Cont.)

Switch	Function
/ASSIGN:dev:log	<p>Assigns a device to your job and a logical name to the device.</p> <p><b>dev:</b> is the device name <b>log</b> is the logical name</p> <p>See Section 1.8.1 for more information about device names.</p> <p>The second argument is optional. This switch can be used more than once to assign more than one device.</p>
/ATTACH: IGNORE or ASK	<p>Sets the ATTACH state of your job. /ATTACH:ASK is the default action. That is, if you log in, detach from the job, and then log in to another job, LOGIN will ask whether you want to ATTACH to the old job or create a new job. (See Example 2.) If you log in using /ATTACH:IGNORE, detach from the job, and log into a new job, LOGIN will not ask if you want to attach to the first job.</p>
/CORE:nP	<p>Tells the system that your job can use a maximum of n 512-word pages. (Refer to the CORE command.)</p>
/CORE:nK	<p>Tells the system that your job can use a maximum of n 1024-word blocks. (Refer to the CORE command.)</p>
/DEFER	<p>Tells the system to defer queueing of spooled output until you log out.</p>
/DEFPROT:nnn	<p>Sets the default protection code for all files created to nnn. (Refer to the SET DEFAULT PROTECTION command.)</p>
/DSKFUL:ERROR	<p>Causes a disk full or quota exceeded error condition to stop output for your job, and passes an error condition to the program. Most programs respond to this condition by printing an error message and returning your job to monitor level without giving you a chance to continue. The default setting is ERROR unless specified as PAUSE.</p>

## SYSTEM COMMANDS AND PROGRAMS

### LOGIN Command (Cont.)

Switch	Function
/DSKFUL:PAUSE	Causes a disk full or quota exceeded error condition to stop output for your job, and suspends execution of the program. The system prints an error message on your terminal and returns control of your job to the monitor. To request assistance from the operator, you can issue the SEND command. To resume execution of your program, issue a CONTINUE command. The program will resume if you did not issue a command that destroys the interrupted program's core image. However, the program will be stopped again if the problem of insufficient disk space or of an exceeded quota has not been corrected. The default setting is /DSKFUL:ERROR.
/DSKPRI:n	<p>Allows privileged users to set the priority for their job's disk operations (data transfers and head positionings). Because the normal timesharing priority is 0, and values range from -3 to +3, the priority n can be higher or lower than standard. The specified priority applies to all disk I/O channels opened whose priority is not explicitly set with a DISK monitor call. (See the <u>TOPS-10 Monitor Calls Manual</u>.)</p> <p>The priority specified in DSKPRI remains in effect until one of the following occurs:</p> <ol style="list-style-type: none"><li>1. A SET DSKPRI command with a different priority is issued.</li><li>2. A KJOB command is issued.</li><li>3. Your program overrides the /DSKPRI switch by issuing a DISK monitor call with a different priority, in an assembly language program.</li></ol> <p>Restrictions are: the privileges required for using this switch are determined by bits 1 and 2 of the privilege word, .GTPRV; these two bits specify an octal number from 0-3; you are always allowed a 0 priority.</p>
/GUIDELINE	Specifies that the numeric value cited in the /PHYSICAL switch is a guideline. This is the default setting for /PHYSICAL.
/LIMIT	Specifies that the numeric value cited in the /PHYSICAL switch is a limit rather than a guideline. If /LIMIT is not issued, the system assumes that the numeric value given for the /PHYSICAL switch is a guideline.

**LOGIN Command (Cont.)**

Switch	Function								
/NAME:name	Associates the given name with your job. This name will appear on output, listings, and on other information output by the system. Enclose the name in quotation marks if it contains non-alphanumeric characters.								
/NEW	Tells the system to add the NEW: directory to your SYS: specification. As a result of this switch, the system, in conducting a search for any program or file, will first check the experimental directory								
/NONEW	Removes the [1,5] directory (NEW:) from your SYS: specification. This is the default setting.								
/NOSCAN	Cancels the /SCAN switch for the directory path.								
/NOSETTY	Tells the system not to change any of your terminal's characteristics as designated by the monitor or in any SET TTY commands you have issued before logging in.								
/NOSTR	Suppresses the printing of SYS:STR.TXT.								
/NOSYS	Removes SYS: structure from your DSK: specification.								
/NOTE:file-spec	Prints the specified file, using the NOTICE.TXT logic. That is, it must be followed by the file name and extension. It is useful for printing a project notice that is being kept in a library area. For example, NOTE:*.TXT[33,437] will print all files with the extension .TXT. If the project number is separated from the programmer number by a slash (/), instead of a comma, the files will be printed only if they were created after you typed the last LOGIN command.								
/NOTICE:arg	Controls printing of NOTICE.TXT and the argument to the /NOTE: switch. The arguments are:								
	<table border="0" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">Argument</th> <th style="text-align: left;">Function</th> </tr> </thead> <tbody> <tr> <td>ALWAYS</td> <td>Always prints notices.</td> </tr> <tr> <td>SOMETIMES</td> <td>Prints notices you have not yet seen.</td> </tr> <tr> <td>NEVER</td> <td>Never prints notices.</td> </tr> </tbody> </table>	Argument	Function	ALWAYS	Always prints notices.	SOMETIMES	Prints notices you have not yet seen.	NEVER	Never prints notices.
Argument	Function								
ALWAYS	Always prints notices.								
SOMETIMES	Prints notices you have not yet seen.								
NEVER	Never prints notices.								
/NOWATCH	Suppresses the printing of incremental job statistics.								

## SYSTEM COMMANDS AND PROGRAMS

### LOGIN Command (Cont.)

Switch	Function
/PASSWORD	Causes LOGIN to ask for a new password after validating your current one. The new password will be stored in ACCT.SYS and will become your permanent password. This feature is specific to certain installations.
/PATH:[dir]	Gives your job a default path for disk I/O. Refer to Section 1.13 for further information.
/PHYSICAL:nP	Specifies the maximum current physical page limit for your job (CPPL; see the <u>TOPS-10 Monitor Calls Manual</u> ). Because P stands for pages, the notation n represents the maximum number of pages. 1P, therefore, would set a limit of one 512-word page. The value that it specifies will be either a PHYSICAL GUIDELINE or a PHYSICAL LIMIT, depending on whether you type /PHYSICAL:nP/GUIDELINE or /PHYSICAL:nP/LIMIT. If you type neither, /GUIDELINE is assumed.
/PHYSICAL:nK	Specifies the number of blocks (K) used as the maximum current physical page limit; 1K would set a limit of one 1024-word block. The value that it specifies will be either a PHYSICAL GUIDELINE or a PHYSICAL LIMIT, depending on whether you type /PHYSICAL:nP/GUIDELINE or /PHYSICAL:nP/LIMIT. If you type neither, /GUIDELINE is assumed.
/QUOTA:str or /QUOTA:(str1,...)	Recomputes disk usage on all specified file structures. If none is specified, usage is recomputed on all file structures in your search list. If a second user logs in under a specific project-programmer number, the /QUOTA switch is ignored.
/RUN:file-spec	Runs the specified program immediately after LOGIN, unless another program has been designated by the system manager.
/SCAN	Sets the /SCAN switch for the directory path. (Refer to SETSRC.)
/SETTTY	Sets your terminal characteristics as specified in the file SYS:TTY.INI. This switch is useful if the INITIA program has not run for your terminal.
/SFDPROT:nnn	Sets the protection of all created SFDs to nnn.



## LOGIN Command (Cont.)

Switch	Function
/SPOOL:dev or /SPOOL:(dev1,...) or /SPOOL:ALL	Adds the specified device(s) to the current list of those spooled for the job. Spooling is the mechanism by which I/O to or from slow-speed devices is simulated on disk. Data temporarily stored on disk can be automatically output on the specified device when it becomes available. These devices can be spooled: the line printer (LPT:), the card punch (CDP:), the card reader (CDR:), the paper-tape punch (PTP:), and the plotter (PLT:).
/STR	Causes all files in the standard system library [1,4] with name ???STR.TXT to be printed on your terminal. Assume, for example, that there is a file on private structure "PR:" called PRLSTR.TXT[1,4] which says: "PR:PAYROLL DEBUG PACK -- NOT REAL DATA". Any user who logged in with /STR (or had /STR in his SWITCH.INI file) would have the warning message printed on his terminal.
/SYS	Adds the SYS: structure to your DSK: specification. Consequently, if a file is not found in the directories in your search list or in your library directory (if /LIB: [proj,prog] has been specified), the system directory [1,4] will then be searched for the file.
/TERMINAL: (arg,arg,...)	Sets the terminal characteristics specified by the arguments. This switch is useful for recording terminal characteristics in your SWITCH.INI file. Refer to Appendix B for information about the SWITCH.INI file. To specify a single argument, type it after the colon. To specify two or more arguments, enclose them in parentheses and separate them with commas.  The arguments to the /TERMINAL switch and their meanings are listed below. All of the arguments except those which take values can be preceded by NO to turn off the function of the argument.
/TIME:n	Sets a central processor time limit of n seconds for a job. When the time limit is reached, the system stops the job and prints a message. A timesharing job can be continued by typing CONTINUE, but unless the time is reset with the SET TIME command, no time limit will be in effect. A batch job cannot be continued.
/UFDPROT:nnn	Sets the protection of all created UFDs to nnn.

**LOGIN Command (Cont.)**

Switch	Function
/VIRTUAL:n	Specifies the current virtual page limit, as represented by n. (For a description of CVPL, see the <u>TOPS-10 Monitor Calls Manual</u> .) In /VIRTUAL:nK and :nP, K represents a block, and P a page; 1K equals 1024 words, and 1P equals 512 words. If you type neither letter, K is assumed. K can be specified within the range 1 to 512P. If you type /VIRTUAL:0, the value of CVPL, as set by the system administrator, is used.
/WATCH: arg,arg...	Duplicates the SET WATCH command, printing messages automatically according to the argument expressed. See the SET WATCH command description for a list of arguments and their uses.

**Arguments to /TERMINAL:**

Argument	Meaning
ALTMODE NOALTMODE	Converts the ASCII characters 175 and 176 to ALTmode (ESCape). If you use NOALTMODE, 175 and 176 regain their original identity as right brace (]) and tilda (~). The default setting is NOALTMODE.
BLANKS NOBLANKS	Prints blank lines during output to the terminal. NOBLANKS is often used on a display terminal to conserve space on the screen. The default setting is BLANKS.
CRLF NOCRLF	Prints an automatic carriage-return/line-feed at the end of the terminal line. The width of this line is set with WIDTH. NOCRLF suppresses the automatic RETURN. The default is CRLF.
DEBREAK NODEBREAK	Notifies the system of the feature that allows the system to lock the keyboard of the terminal. This feature is used for Model 2741 terminals. NODEBREAK notifies the system that the terminal cannot be locked.
DEFER NODEFER	Suppresses echoing of the characters you type until output to the terminal is finished. For example, when the system is sending output to your terminal and you type another command, the system will echo the characters as it is printing output on your terminal. DEFER holds the characters you type until the output is finished. NODEFER is the default setting. For video terminals, it is recommended that you set DEFER.

## SYSTEM COMMANDS AND PROGRAMS

### LOGIN Command (Cont.)

Argument	Meaning
DISPLAY NODISPLAY	Informs the system that you have a display terminal. Your programs can use this information when sending output to your terminal. NO DISPLAY turns off the DISPLAY function.
ECHO NOECHO	Prints the characters you type on your terminal. This puts your terminal in full duplex mode. NOECHO puts your terminal in half duplex mode. That is, the characters you type are not printed on your terminal.
FILL:n NOFILL	Controls the filler class of the terminal. This effects the output of filler characters. NOFILL is the equivalent to FILL:0.
FORM NOFORM	Sends eight line-feeds for every FORM character, and four line-feeds for each vertical tab. NOFORM does not send the line-feeds.
GAG NOGAG	Suppresses any messages sent by the SEND command when your job is in user mode. This does not effect messages from the operator. NOGAG allows you to receive messages at any stage of your job.
LC NOLC	Allows the system to print lowercase characters on your terminal when echoing characters from your terminal. This argument is used on terminals that have lowercase ability but are not printing lowercase characters. NOLC makes the system translate all input characters to uppercase as they are transmitted.
PAGE NOPAGE	Allows you to use CTRL/S and CTRL/Q to control the output to your terminal. These control characters are described in Section 1.5. NOPAGE allows the output to your terminal to keep printing with no interruption.
PAGESIZE:n	Sets the number of lines per terminal page. This makes the system stop the output to your terminal after n lines, ring your terminal bell, and continue when you type CTRL/Q. If you specify 0 as the page size, the output will not stop automatically, but it will stop when you type CTRL/S. Refer to Section 1.5.
RTCOMP NORTCOMP	Controls the function of CTRL/R and CTRL/T. NORTCOMP makes the control characters function as they are described in Section 1.5. RTCOMP prevents them from functioning this way, which is useful when you plan to run a program that uses CTRL/R and CTRL/T for other purposes.

**LOGIN Command (Cont.)**

Argument	Meaning																
RCVSPEED:nnnn	Sets the speed at which your terminal will receive characters to n baud.																
SPEED:nnnn	Sets the receiving and transmitting speed of your terminal to n baud.																
TABS NOTABS	Informs the system that your terminal has tab stops. NOTABS informs the system that your terminal does not have tab stops. The system will then simulate tab stops for your terminal.																
TAPE NOTAPE	Informs the system that your terminal has paper tape output. This changes the function of CTRL/S and CTRL/Q to control the paper tape. NOTAPE restores the function of CTRL/S and CTRL/Q. Refer to Section 1.5.																
TIDY NOTIDY	Specifies that each character occupy one character space. This is useful on Model 2741 terminals, where the [ character usually appears as ^<. In TIDY mode, [ appears as <. NOTIDY turns this function off.																
TYPE:nnnn	Specifies the type of terminal. This informs the system that your terminal is of the type nnnn. This sets some characteristics automatically. These are shown in Table 2-2. The value nnnn can be any of the following: <table border="1" data-bbox="727 1150 1136 1260"> <tbody> <tr> <td>DAS21</td> <td>LA120</td> <td>LA34</td> <td>LA36</td> </tr> <tr> <td>LA38</td> <td>VT100</td> <td>VT61</td> <td>VT05</td> </tr> <tr> <td>VT06</td> <td>VT50</td> <td>VT52</td> <td>TTY33</td> </tr> <tr> <td>TTY35</td> <td>LA30</td> <td>2741</td> <td>TTY</td> </tr> </tbody> </table>	DAS21	LA120	LA34	LA36	LA38	VT100	VT61	VT05	VT06	VT50	VT52	TTY33	TTY35	LA30	2741	TTY
DAS21	LA120	LA34	LA36														
LA38	VT100	VT61	VT05														
VT06	VT50	VT52	TTY33														
TTY35	LA30	2741	TTY														
UC NOUC	Tells the system to translate all the characters that it receives from your terminal to upper case. NOUC does not translate the characters.																
WIDTH:n	Sets the width of the terminal screen to n. Used in conjunction with CRLF, this controls the automatic RETURNS that are output at monitor level.																

**Characteristics**

The LOGIN command:

Returns your terminal to monitor level or starts a program if specified in ACCT.SYS.

Runs the LOGIN program.

## SYSTEM COMMANDS AND PROGRAMS

### LOGIN Command (Cont.)

#### Associated Messages

If you are already logged in when you issue the LOGIN command, the monitor prints:

```
?PLEASE KJOB OR DETACH
```

If the system is running the maximum number of jobs it can handle, you will not be able to log in. When you issue a LOGIN command, you will receive the message:

```
?JOB CAPACITY EXCEEDED
```

In this event, wait a few minutes, then try again.

#### Examples

1. The following is the procedure to gain access to the system:

```
.LOGIN(RET)  
JOB 29 RZ373B KL #1026/1042 TTY220  
#27,5434(RET)
```

LOGIN prints your assigned job number (job number 29), followed by monitor name, version number, and terminal line number. If you do not type your project-programmer number on the same line as the LOGIN command, LOGIN prints a number sign (#) indicating that you should type your project-programmer number.

```
PASSWORD:(RET)
```

The system prompts you for your password. You type your password followed by a carriage-return. To maintain password security, the monitor does not echo your password. On terminals with local-copy (see the TOPS-10 Monitor Calls Manual), a mask is printed to make your password unreadable.

```
14:02 10-SEP-80 MON
```

If your 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.

SYSTEM COMMANDS AND PROGRAMS

**LOGIN Command (Cont.)**

2. The following example shows the LOGIN procedure after you detach from your first job.

Log in just as you did the first time.

```
.LOG 27,5434 (RET)
JOB 33 RZ373B KL #1026/1042 TTY220
PASSWORD (RET)
```

```
OTHER JOBS DETACHED WITH SAME PPN:29
TYPE JOB NUMBER TO ATTACH OR CARRIAGE-RETURN TO LOGIN NEW
JOB: (RET)
```

Type carriage-return to create a new job.

```
[LGNJSP OTHER JOBS SAME PPN:29]
14:10 10-SEP-80 MON
```

**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, the system prints a warning message. If you continue in TECO, the system program supersedes the file. If you type two CTRL/Cs to leave TECO, the program does not destroy the file. (See the TECO manual in the TOPS-10 Software Notebooks.)

**Format**

MAKE dev:file.ext[directory]

Where: **dev:** is the device or file structure name on which the system creates the file. If you omit it, the system assumes DSK:.

**file.ext** is any legal file name and file name extension. The file name is required; the file name extension is optional.

**[proj,prog]** is the directory area in which the system creates the file. If you omit this argument, the system assumes your default directory area (that is, your project-programmer number). Note that the default directory can be an SFD or a UFD.

You can pass switches to TECO if you precede each switch with a slash in the MAKE command string.

**Characteristics**

The MAKE command:

Places your terminal at user level.

Destroys your core image.

**Example**

Create a file named TEXT3.MAI.

```
.MAKE TEXT3.MAI 
```

```
*EX  
```

Exit from TECO.

Use DIRECTORY to see the file.

```
.DIR TEXT3 
```

```
TEXT3 MAI 1 <055> 10-SEP-80 DSKC:[27,5434]
```

.

**MERGE Command****Function**

The MERGE command combines the low segment of an executable program in the specified file with the program that is currently in memory. The MERGE command, like the GET command, does not start execution of the program. MERGE is used to load page fault handlers and DDT.

**Format**

```
MERGE dev:file.ext[directory] core
```

Where: **dev:** is the logical or name of the device containing the program you want to merge into core. The default device is DSK:.

**file.ext** is the name of the file containing the program you want to merge into core. You must specify the file name. The file extension defaults to .EXE.

**[directory]** is the directory name, required only if the core image file is located in a disk area other than yours. The default is the directory area that you logged in to.

**core** is the amount of core to be assigned to the merged file if different than the minimum amount of core needed to load the program. The default is the minimum amount of core needed to load the program.

**Characteristics**

The MERGE command:

Places your terminal at user level.

Requires LOGIN.

Changes your core image.

**Example**

Type two programs.

```
.TY FIRST.MAC (RET)
FIRST:  RESET
        JRST   700000
        END    FIRST

.TY SECOND.MAC (RET)
        .PSECT FOO,700000
SECOND: OUTSTR [ASCIZ/
EXECUTION AT 700000
/]
        EXIT
        END
```



**MERGE Command (Cont.)**

Compile the programs.

```
.COMPILE FIRST.MAC (RET)
MACRO: .MAIN
```

EXIT

```
.COMPILE SECOND.MAC (RET)
MACRO: .MAIN
```

EXIT

Run LINK and save the files.

```
.R LINK (RET)
```

```
*FIRST/SAVE=FIRST/GO (RET)
```

EXIT

```
.R LINK (RET)
```

```
*SECOND/SAVE=SECOND/GO (RET)
```

EXIT

Use GET to bring the first program into memory.

```
.GET FIRST (RET)
JOB SETUP
```

Use MERGE to bring the second program into memory.

```
.MERGE SECOND (RET)
SECOND MERGED
```

Start execution of the programs.

```
.START (RET)
```

EXECUTION AT 700000

EXIT

After a successful execution, examine the memory area. Note that address 777 is occupied.

```
.E 777 (RET)
000777/ 000000 000000 .
```

Address 1000 does not exist.

```
.E 1000 (RET)
?OUT OF BOUNDS
```

SYSTEM COMMANDS AND PROGRAMS

**MERGE Command (Cont.)**

Address 700000 is occupied.

.E 700000 (RET)  
700000/ 051140 700002 .

Address 701000 does not exist.

.E 701000 (RET)  
?OUT OF BOUNDS

## MIC Commands

### Function

MIC allows you to create a new command by writing any desired sequence of monitor-mode and user-mode commands in a disk file. MIC commands are discussed in the file DOC:MICV2.DOC, on the system. MIC is not a supported product.

You may include any of the following commands in your MIC command file, along with any number of monitor commands.

### Commands

- |                         |  |
|-------------------------|--|
| BACKTO                  | Specifies a 1- to 6- character label at which MIC processing is to resume. Note that labels in MIC command files are terminated by :: and must be at the beginning of a line.  |
| ERROR/<br>NOERROR       | Specifies a character that is used to denote an error condition when displayed at the beginning of a line.   |
| GOTO                    | Specifies a 1- to 6- character label at which MIC processing is to resume. Note that labels in MIC command files are terminated by :: and must be at the beginning of a line.  |
| IF                      | Conditionally processes a monitor command.   |
| LET                     | Allows you to change the value of any user parameters in a MIC file. For example, LET C= "THIS IS C" substitutes the text within double quotes as the value of C.  |
| MIC Arg                 | Allows you to ABORT, SUSPEND, CANCEL, or EXIT from MIC processing.   |
| OPERATOR/<br>NOOPERATOR | Specifies one ASCII character to be treated as introducing a line that requires user attention. For example, if the operator character is output in column 1, MIC suspends output and displays [BREAK]. After this, type the requested information. Then type CTRL/P (PROCEED), which will continue the processing of the command file.  |
| SILENCE/<br>REVIVE      | Suppresses and resumes output to the terminal. You can type another monitor command on the same line as the SILENCE/REVIVE command. For example,<br><br>.REVIVE.DIR<br><br>causes terminal output to be revived after the .DIR command has been typed. Therefore, only the output from the .DIR command will be printed on the terminal. |
| WHENEVER/<br>ON         | Changes the default action whenever a particular action or event occurs. ON and WHENEVER are equivalent commands.  |

## MIC Commands (Cont.)

## Example

```

;A MIC DEMO OF THE ERROR COMMAND
;A MACRO TO COMPARE TWO FILES AND DELETE DSK COPY IF THEY
;ARE THE SAME
.
.TYPE CHECK.MIC (RET)

.ERROR ?
.R FILCOM
*TTY:/Q='A.'B
.DELETE 'A

.
;A TYPICAL CALL - CHECK IF DSK:C.MIC IS A COPY OF SYS:C.MIC
.DO CHECK C.MIC,SYS:(RET)

.
.ERROR ?

.R FILCOM

*TTY:/Q=C.MIC,SYS:
NO DIFFERENCES ENCOUNTERED

*^C

.DELETE C.MIC
FILES DELETED
C.MIC
01 BLOCKS FREED
;CREATE AN INCORRECT COPY OF C.MIC WITH PIP.
.R PIP (RET)

*C.MIC=TTY:
WRONG FILE
^Z
*^C
;AND USE C.MIC TO CHECK IT
.DO CHECK C.MIC,SYS:(RET)

.
.ERROR ?
.R FILCOM (RET)

*TTY:/Q=C.MIC,SYS:(RET)
FILE 1) DSK:C.MIC      CREATED: 1349  21-JAN-1979
FILE 2) SYS:C.MIC     CREATED: 1202  03-OCT-1978

?FILES ARE DIFFERENT

*[ABORT ON ERROR]
^C
.

```

## MOUNT Command

### Function

The MOUNT command requests assignment of a device from the operator. The MOUNT command requests operator action when necessary. For example, if you request a tape drive and all drives are in use, the operator can free one for you. With the MOUNT command you can request devices from the restricted pool of resources.

The MOUNT command gives the operator control over the assignment of devices on the system. When you request a device with this command, the operator has the option of selecting a specific unit (for example, DTA5), deferring your request, or canceling your request. The operator can also mount the medium for the requested unit if it is sufficiently identified (for example, a magnetic tape on a tape drive).

When you use MOUNT to access a file structure, you can mount the structure on a specific drive, place the structure in your job search list, and wait for the operator to mount the disk pack.

Each file structure can have an administrative file, QUOTA.SYS, which contains a list of quotas for all users allowed to access the structure. (Quotas for the structures in your default search list are contained in the file SYS:AUXACC.SYS.) When the operator mounts a file structure, the system creates a UFD for you if you have an entry in the file structure's QUOTA.SYS or in SYS:AUXACC.SYS.

The MOUNT command runs the UMOUNT program. 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. OMOUNT interacts with the operator. UMOUNT waits for OMOUNT to finish, unless you have specified /NOWAIT or you type two CTRL/Cs. When you type two CTRL/Cs, you do not receive a message when the operator has mounted the device, but you can use the /CHECK switch to see if your request is in the MOUNT queue. (See the examples.)

To insure validity of any tape error analysis, you should use MOUNT and DISMOUNT to acquire and release magnetic tape units. This provides the basis for all media-related error reporting.

### Format

MOUNT dev:logical-name/switches(drives)

Where: **dev:** is any physical device name, previously assigned logical name, or file structure name. Device names are discussed in Section 1.8.1.

**logical-name** is any set of six or less alphanumeric characters. The logical name can be a previously assigned name. If the previous request has been processed, MOUNT transfers the logical name from the previously assigned device to the device in this MOUNT command. If the previous request has not been processed at the time you specify the logical name again, then MOUNT creates a unique logical name of the form "Mnnnnn."

## SYSTEM COMMANDS AND PROGRAMS

### MOUNT Command (Cont.)

It is recommended that you supply a logical name with any non-disk MOUNT request because the logical name is the only way for you to refer to the device that was assigned to your job.

**/switches** further describe the requested action. Switches are optional unless otherwise indicated in the individual switch descriptions. The system processes the switches in the order that you type them. Therefore, if there are any conflicting switches, the system uses the last one in the command string. However, if you specify both **/WAIT** and **/NOWAIT** in the string, the command cannot be processed and your job receives a fatal error. You can abbreviate the switch names to as few characters as are necessary to make the switch unique. The switches to MOUNT are described below.

**(drives)** are the physical drives on which the units are to be mounted. A drive argument can 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 drives that are not specified must be represented by commas. For example, (**,, RP03**). Unspecified trailing drives can be omitted. If you do not specify a drive, the system finds an available drive of the proper type. If you specify a drive incorrectly, or if the drive is unavailable, you are informed and the requests are then queued.

You can use the following switches in the command string.

Switch	Function
<b>/ACTIVE</b>	Mounts the specified file structure in your active search list. Refer to the SETSRC program. <b>/ACTIVE</b> is the default function.
<b>/CHECK</b>	Prints a list of your job's pending MOUNT requests on your terminal. This switch is intended to be used alone with the MOUNT command. If a device specification or other switches appear in the command string with the <b>/CHECK</b> switch, the system ignores them.
NOTE	
<b>/C</b> is a valid abbreviation of <b>/CHECK</b> . Therefore, you must type at least <b>/CR</b> when you desire <b>/CREATE</b> .	
<b>/CREATE</b>	Permits the creation of files on the specified file structure. This is the default function. This switch implies <b>/ACTIVE</b> . Refer to the SETSRC program.
<b>/NOCREATE</b>	Prevents the creation of files on the specified structure unless the structure is an output device. This switch implies <b>/ACTIVE</b> . Refer to the SETSRC program.

## SYSTEM COMMANDS AND PROGRAMS

### MOUNT Command (Cont.)

Switch	Function
/HELP	Prints a brief description of the MOUNT command on your terminal. This switch is intended to be used alone with the MOUNT command. If a device specification or other switches appear in the command string with the /HELP switch, they are ignored.
/MULTI	Allows multiple access to the file structure. This is the complement to /SINGLE. /MULTI is the default function.
/PASSIVE	Adds the file structure to your passive search list. The structure is not considered part of generic DSK: and is not automatically searched. Refer to the SETSRC program.
/PAUSE	Notifies you before queuing the mount request to the operator. This gives you the opportunity to cancel the request.
/REELID:name	Specifies the identification of the tape reel. The identification (name) is a set of 6 or less alphanumeric characters that uniquely identify the tape. This identification is included in any system-generated error or status reports about the drive while the reel is mounted. This switch is required when mounting magnetic tapes. In order to associate the error reports with the reel on which the errors occurred, it is recommended that a label bearing the reel identification be attached to the tape reel. Refer to the /VOLID switch description.
/ROONLY	Specifies that the device is read only. This is the same as /WLOCK.
/SHOVE	Forces the mount request to the operator even if SCHED 400 is set. Refer to the SCHED command.
/NOSHOVE	Does not force the request to the operator if SCHED 400 is set. Refer to the SCHED command. This is the default function.
/SINGLE	Prevents access to this structure by other users. You must have the same project number as the owner of the pack to be allowed single access. This switch applies only to disk mount requests.

## SYSTEM COMMANDS AND PROGRAMS

### MOUNT Command (Cont.)

Switch	Function
/VID:text	<p>Specifies the visual identification of the volume. The identification (text) is passed to the operator. This switch is required in MOUNT requests for DECTapes and for all input-only devices.</p> <p>The text can be up to 50 characters in length. For DECTapes, the text is the DECTape number, for example, /VID:1326. If the tape is not numbered or if you do not want to have the number checked, you must describe the tape by enclosing the text in single or double quotation marks. For example:</p> <pre>MOUNT DTA:/VID:"FAILSAFE FOR [13,2652]"</pre> <p>For input-only devices, the text should be a description of the input. For example:</p> <pre>MOUNT CDR:/VID:"PLEASE LOAD THE BLUE CARD-DECK:'COBOL SOURCE'"</pre> <p>Unquoted strings can contain only alphanumeric characters, periods, and hyphens. In quoted strings, the quotation marks are included in the 50 characters.</p>
/WAIT	<p>Waits before continuing your job when operator intervention is required. When this switch is used in a command with a device specification and other switches, your job will wait for the request to be completed. When this switch is used alone (MOUNT/WAIT), your job will not continue until all of your pending requests have been processed. /WAIT is the default condition. /W is a valid abbreviation of /WAIT.</p>
/NOWAIT	<p>Continues the job immediately after queuing the MOUNT request.</p>
/WENABL	<p>Allows you to write files onto the specified device. This is the default function for file structures.</p>
/WLOCK	<p>Prevents you from writing files onto the device. The monitor will not update the access date. If you specify /SINGLE with this switch, the operator can set a write lock. /WLOCK is the default for magnetic tape and DECTape.</p>
/WRITE:NO	<p>Prevents you from writing onto the device. Same as /WLOCK.</p>
/WRITE:YES	<p>Permits you to write files onto the device. Same as /WENABLE.</p>



## MOUNT Command (Cont.)

## Characteristics

The MOUNT command:

Places your terminal at user level.

Destroys your job's core image.

## Examples

1. The MOUNT command requests that the operator mount the magnetic tape identified as "Thursday's FAILSA for DSKA." The /REELID switch specifies that the tape is to be identified as reel number 456 in all error reports. This command specifies that the logical name FAILSA be associated with the drive on which the reel is mounted.

```
.MOUNT MTA:FAILSA/REELID:456/VID:"THURSDAY'S TAPE FOR DSKA" (RET)
REQUEST QUEUED
WAITING...2 ^C'S TO EXIT
^C
^C
.
```

2. The following command asks the operator to mount a magnetic tape with a REELID of 123 and the external label "PAYROLL MASTER" on MTA1:. The operator was unable to use MTA1: so he selected MTB1: instead. However, OMOUNT assigned the logical name MTA1: to the physical device MTB1:. This feature is especially useful in batch control files and allows you to continue to refer to the tape as MTA1:.

```
.MOUNT MTA1:/REELID:123/VID:"PAYROLL MASTER" (RET)
REQUEST QUEUED
WAITING...2 ^C'S TO EXIT
MTA MOUNTED, MTB101 USED
```

```
.SYSTAT (RET)
24 10,470244 TTY11 SYSTAT 12+SPY RN 23 $
$ MEANS EXECUTE ONLY
```

```
BUSY DEVICES:
DEVICE      JOB      WHY      LOGICAL

MTB101      24      AS      MTA101
```

3. This MOUNT command asks the operator to mount the file structure BLKC:. You want to return to monitor command level after the request is queued, without waiting for the operator to service the request.

```
.MOUNT BLKC:/NOWAIT (RET)
REQUEST QUEUED TO MOUNT UNITS
```

## SYSTEM COMMANDS AND PROGRAMS

### MOUNT Command (Cont.)

4. Request that the operator mount the DECTape labeled 5678. You are notified that the DECTape is mounted on DTA1:.

```
.MOUNT DTA:/VID:5678(RET)
REQUEST QUEUED
WAITING...2 ^C'S TO EXIT
DTA MOUNTED, DTA101 USED
```

5. The response from the MOUNT/CHECK command indicates your requests that have not been processed. The mount request for DECTape 1234 is still pending. /AFTER tells you that the request has been deferred until 17:20 on January 20, 1980. The last line tells you that the total number of MOUNT requests pending for all users is one.

```
.MOUNT/CHECK(RET)
1. M JOB25 TTY105 10,444511 1 MOUNT DTA M44321/V:1234
/WL/AFTER:20-JAN-80:17:20
1 COMMAND IN QUEUE
```

## NETWORK Command

### Function

The NETWORK command prints information about the systems in a network environment. Your terminal must be attached to one of the systems in the network to use this command. You can use the NETWORK command to obtain information about the other systems in the network, and about the device configuration of each system in the network.

You can specify the node-name or node-number of systems in the network, to limit the output to specific nodes. You can use switches to change the output, and to specify the configuration characteristics that you want to see.

NETWORK reads your SWITCH.INI file, and accepts indirect commands. The indirect command file should be formatted as follows:

```
node-list/switch/switch...
```

Indirect command files are discussed in Appendix C.

### Formats

```
NETWORK node-list/switch
```

```
R NETWORK (RET)
*node-list
```

Where: **node-list** is the node-names or node-numbers for which you want information. If you do not specify the node-list, system information is output for every node in the network. Separate the node-names or node-numbers by commas.

\* is the NETWORK prompt.

You can use the wildcard construction in the command string. The node name, or any part of the node name, can be replaced by \* or ? to represent groups of nodes. Wildcards are discussed in Section 1.11.

The switches to NETWORK allow you to specify:

- The format of the output.
- The configurations that you want to examine.

The switches to the NETWORK command follow:

#### Output Control Switches

Switch	Function
/BRIEF	Suppresses the output of a list of devices for each node. This is the default if you do not specify a node-list in the command string.
/ERROR	Prints all error messages. This is the default.

## SYSTEM COMMANDS AND PROGRAMS

### NETWORK Command (Cont.)

Switch	Function
/HELP	Prints a description of the NETWORK command.
/NOBRIEF	Prints a list of the devices on each node. This is the default if you specify a node-list.
/NOERROR	Suppresses the output of error messages.
/NOHELP	Suppresses the output of HELP text.
/NOSILENCE	Prints all information. This is the default.
/SILENCE	Prints only error messages.
/SORT	Sorts the nodes according to node number.

#### Node Selection Switches

Switch	Function
/CDP	Prints only nodes that have card-punch devices.
/CDR	Prints only nodes that have card readers.
/DTA	Prints only nodes that have DECTape units.
/LPT	Prints only nodes that have line printers.
/MCR	Prints only nodes that have command interpreters (monitors). MCR signifies that the node is a host system.
/MTA	Prints only nodes that have magnetic tape units.
/NOCDP	Prints only nodes that do not have card-punch devices.
/NOCDR	Prints only nodes that do not have card readers.
/NODTA	Prints only nodes that do not have DECTape units.
/NOLPT	Prints only nodes that do not have line printers.
/NOMCR	Prints only nodes that do not have command interpreters.
/NOMTA	Prints only nodes that do not have magnetic tape units.
/NOPLT	Prints only nodes that do not have plotters.
/NOPTP	Prints only nodes that do not have paper tape punch units.
/NOPTR	Prints only nodes that do not have paper tape readers.

**NETWORK Command (Cont.)**

Switch	Function
/NORDA	Prints only nodes that do not have RDX devices. RDX devices control lines from high-speed input (block-mode) terminals.
/NOTSK	Prints only nodes that do not have TSK: devices.
/NOTTY	Prints only nodes that do not support terminals.
/TYPE:DNxxx	Prints only nodes of the specified type. This does not output host nodes. However, if you use /TYPE: with no value, the host nodes are printed.
/PLT	Prints only nodes that have plotters.
/PTP	Prints only nodes that have paper tape punch units.
/PTR	Prints only nodes that have paper-tape readers.
/RDA	Prints only nodes that have RDX devices. RDX devices control lines from high-speed input (block-mode) terminals.
/TSK	Prints only nodes that have TSK: devices.
/TTY	Prints only nodes that support terminals.

**Network Topology**

The lines between nodes in the network represent routes from one node to the next. Two nodes can be connected by a variety of routes, including direct lines, lines with intercepting nodes, and node-to-node. Each line may be characterized by its "cost." The cost of a line is an arbitrary number used to control the route between one node and another. The greater the cost of a line, the less desirable that route may be in travelling from one node to another. You use the SET HOST and LOCATE commands to connect your terminal to another node. The system uses the cost of the available routes to determine the most efficient route to take from one node to another.

Using the NETWORK command with the /TOPOLOGY switch, you can determine the lines and routes that connect one node to another node. The output shows the name of each node in the network. Following the node name, the names of the nodes that are on a direct line are shown. By tracing the direct lines that are output by the NETWORK command, you can determine the various routes you may take to gain access to another node. The following switches allow you to determine the cost of each line. The output of these switches and the use of topology are illustrated in Example 4.

## NETWORK Command (Cont.)

Switch	Function
/COST	Shows the cost of the line from the subject node. The subject node is the first system in the output, the cost of the line is in parentheses after the node name. This output is the default when you use the /TOPOLOGY switch.
/NOCOST	Used with the /TOPOLOGY switch, shows the nodes on each direct line, but does not show their costs.
/TOPOLOGY	Prints the node name, the nodes to which it is connected, and the cost of each line.

## Associated Messages

The output from the network command is in one of the following formats.

When you do not specify a node-list:

```
NODE node-name (node-number) system-name system-generation-date
```

When you specify a node-list:

```
NODE node-name (node-number) system-name system-generation-date
device[number-of-devices] device[number-of-devices]
device[number-of-devices]...
```

## Characteristics

The NETWORK command:

Runs the NETWORK program.

Does not require that you be logged in.

## Examples

- The following example shows the NETWORK command and output, when the node-list is omitted. There are 5 nodes on the network.

```
.NETWORK (RET)
NODE    KL1026 (26)    RZ34B KL #1026/1042  02-13-80
NODE    KS4101 (76)    RS034A KS - 4101 02-13-80
NODE    TWINKY (71)    TWINKY 700/376B 02-06-80
NODE    WIZARD (123)   DN81 V20C(D)
NODE    STARS (65)    DN20 V21(131) 25-DEC-79
```

SYSTEM COMMANDS AND PROGRAMS

NETWORK Command (Cont.)

2. The following example shows the NETWORK command when the node-list is specified. One node-number is valid, the other is invalid.

```
.NETWORK 26,41 (RET)

%NWKNNN NODE 41 NOT IN NETWORK
NODE     KL1026 (26)      RZ34B KL#1026/1042 02-13-80
        CDP[1] TSK[1] DTA[8] MTA[15] PLT[1] PTP[2] PTR[2]
        LPT[5] CDR[2] TTY[137] MCR[1]
```

3. The following example shows the use of NETWORK with the /NOTSK switch.

```
.NETWORK/NOTSK (RET)

NODE     WIZARD      (123)   DN81 V20C(D)
NODE     STARS       (65)    DN20 V21(131) 25-DEC-79
```

4. The following example shows the use of the NETWORK command with the /TOPOLOGY switch. The connected node (the node that is running your job) is KL1026. 26 has a direct line to node 32 (ENCORE), node 31 (NOVA), node 34 (JINX), node 33 (DWARF), and node 27 (NEXT). Each of these lines costs 10. Therefore, they are equally acceptable links. Note that NEXT (node 27) has a direct line to KL1026 (node 26), but this line costs 64, and is therefore less acceptable than the line which connects KL1026 to NEXT. The nodes NCC (node 6) and 2102 (node 20) have no direct lines to another node.

```
.NETWORK/TOPOLOGY (RET)

NODE     KL1026 (26)   32(10)  31(10)  34(10)  33(10)  27(10)
NODE     CTCH22 (22)   76(8)
NODE     KS4101 (76)   22(10)  27(10)
NODE     NCC     (6)    NONE
NODE     SOFDCP (75)   71(8)   6(8)
NODE     TWINKY (71)   75(10)  123(10)
NODE     WIZARD (123)  71(8)   27(8)
NODE     2102   (20)   NONE
NODE     NEXT   (27)   26(64)  123(16) 76(16)
NODE     DWARF (33)   26(64)
NODE     ENCORE (32)  26(64)
NODE     JINX   (34)   26(64)
NODE     NOVA   (31)   26(64)  20(8)
```

## NODE Command

### Function

The NODE command prints network configuration information. If you do not specify a node-id, the system displays information about the node to which your terminal is connected. If you specify a node-id, the system displays system configuration information for the node(s) you specified.

### NOTE

You should use the NETWORK command instead of the NODE command, because NODE will be discontinued in future monitor versions. NODE is described here for the sake of compatibility.

To use some commands (for example, SET HOST), you must know if a node has a command interpreter. The NODE command displays the mnemonic MCR as a device on each node that has a command interpreter.

### Format

NODE node-id

Where: **node-id** is a node identifier of a node in the network. If you specify 0 as the node-id, the host to which your terminal is connected is printed. That is, the system prints the command interpreter of your job. If you do not specify a node-id, the subject of the output is the node to which your terminal is connected.

### Characteristics

The NODE command:

Leaves your terminal at monitor level.

Does not destroy your core image.

### Associated Messages

When you specify a node-id, the information for each node is displayed in the following format:

```
NODE node-name (node-num) software-id creation-date
      device[number-of-devices] device[number-of-devices]...
```

Where node-name is the name of the node, node-num is the node's number, software-id is the name and version of the monitor on that node, and creation-date is the date of the monitor generation.

The second line of output lists each kind of device on the node, and the number of devices.



**NODE Command (Cont.)**

**Examples**

1. The following example shows the command interpreter to which your terminal is connected.

```
.NODE 0 RET
NODE          KL1026(26)      RZ357A KL #1024/1042 06-13-79
                MCR[1] TTY[137] CDR[2] LPT[5] PTR[2] PTP[2] PLT[1]
```

2. Print the information about the node to which your terminal is connected.

```
.NODE RET
NODE DALLAS(3)
TTY[33],CDR[1],LPT[1]
```

3. Print the node information of node number 3.

```
.NODE 3 RET
NODE DALLAS(3)
TTY[33],CDR[1],LPT[1]
```

4. Print the node information for node SCYLLA. The message indicates that node SCYLLA is not on line.

```
.NODE SCYLLA RET
?UNDEFINED NETWORK NODE
```

**NSAVE Command**  
**NSSAVE Command**

**Function**

The NSAVE and NSSAVE commands write a file of your core area on the specified device. The NSAVE command creates non-sharable .EXE files, and the NSSAVE command creates sharable .EXE files.

These commands duplicate the SAVE and SSAVE commands. NSAVE and NSSAVE are used to specify that the .EXE format be used in the core image file.

## SYSTEM COMMANDS AND PROGRAMS

### OPSER Program

#### Function

The OPSER program allows you to control up to 14 subjobs from your terminal. OPSER acts as the supervisor of the subjobs by allowing monitor-level or user-level commands to be passed to all or to selected subjobs. Output from the various subjobs can be retrieved by OPSER.

The subjobs of OPSER run on pseudo-terminals (PTYs) and all initializations of the pseudo-terminals are performed by OPSER. You must provide the subjob name and either an OPSER-provided subjob number or a user-assigned name. System programs, as well as programs you write, that require a dedicated terminal can be run as subjobs of OPSER. By running jobs on pseudo-terminals, OPSER is able to maintain an I/O link between you and the running jobs.

OPSER is used by the system operator to control the separate components of the batch system. However, all users can use this program. (See the OPSER specification in the TOPS-10 Software Notebooks for more information on OPSER.)

#### Format

R OPSER

OPSER prompts by printing an asterisk if no subjobs are in use or if subjobs are waiting for operator action. OPSER responds with an exclamation point when a subjob is running. You can enter commands whenever OPSER is operating. Each command must be preceded by a colon and may be abbreviated to a unique set of characters.

#### OPSER Commands

Command	Function
:AUTO/hh:mm	Processes the specified file as an automatic startup file. The execution of the file is terminated by the end of the file or by input from the terminal. /AUTO: is the way that 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 comments with a semicolon. For example,  :SLOGIN ;LOGIN A NEW SUBJOB
:AUTO/+>hhmm	Processes the AUTO file at the next occurrence of hhmm.
:AUTO/->hhmm	Processes the AUTO file if time is later than hhmm.
:CLOSE	Closes the log file without opening a new one.

SYSTEM COMMANDS AND PROGRAMS

OPSER Program (Cont.)

Command	Function
:CONTINUE	Continues processing the AUTO file after it has been interrupted by a CTRL/C. This allows you to gain control of a subjob during auto file processing.
:CURRENT	Prints the name of the current subjob, if defined; otherwise, prints the number of the current subjob. Output from another subjob does not affect the definition of the current subjob.
:DAYTIME	Prints the current date and time.
:DEFINE xxx=n	Associates the symbol xxx as the name for subjob n. The symbol B is reserved for the subjob running BATCON.
:DEVICE dev:log:n	Assigns the device (dev:), and logical name (log) to subjob n. The logical name is optional, but you must include a null field if the logical name is omitted, for example, :DEVICE CDR::3. The REENTER command will abort the request.
:ERROR n,m,p	Prints only error messages. (That is, ignores nonerror messages from subjob n.) Printing is resumed with the :REVIVE command.
:EXIT	Exits to the monitor if subjobs are not in use; otherwise, give a list of those that are running. This should be used instead of CTRL/C, because :EXIT does not return your job to monitor level if there are any active subjobs. (Also refer to :MONITOR.)
:FREE	Prints the first free subjob number.
:HELP	Prints a text that briefly explains the commands.
:JCONT n	Continues the specified currently halted job.
:KILL n,m,p	Kills the specified subjobs. This is identical to :KJOB.
:KJOB n,m,p	Kills the specified subjobs, saving all files.
:LOGIN proj,prog	Logs in a new subjob. If no project-programmer number is typed, OPSER assumes your project-programmer number.

## SYSTEM COMMANDS AND PROGRAMS

### OPSER Program (Cont.)

Command	Function
:MONITOR	Exits to the monitor, even if subjobs are running. (Also refer to :EXIT.)
:MSGLVL 0	Causes the response to the :WHAT command to include the JOBSTS bits.
:MSGLVL 1	Causes the response to :WHAT command to eliminate the JOBSTS bits. This is the default.
:QUEUE <line>	Initiates the first free subjob and sends the typed-in line to the system queue manager.
:RESOURCES	Prints the list of the available system resources.
:REVIVE n	Resumes normal echoing of output from subjob n (that is, clears the effects of :SILENCE, :TSILENCE, and :ERROR for subjob n).
:SCHED	Prints the schedule bits as set by the operator.

Bit	Meaning
0	Regular timesharing.
1	No further LOGINs 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.
1000	No auto down-line load of nodes.

:SEND	Simulates the SEND command, sending a line of text to the operator's terminal.
:SILENCE n	Suppresses all output from subjob n.
:SLOGIN proj,prog	Logs one subjob in but suppresses its response. If you omit the project-programmer number, OPSER uses yours.

## SYSTEM COMMANDS AND PROGRAMS

### OPSER Program (Cont.)

Command	Function
:STOP n	Puts the specified subjob at monitor level. This is equivalent to typing two CTRL/Cs in interactive mode.
:SYSTAT x	Runs SYSTAT with argument x over the first free subjob. The argument can be any of those listed under the SYSTAT command. The argument is optional.
:TIME	Prints the total running time since the last :TIME command, followed by the integrated product of running time and core size.
:TLOG file-spec	Creates a log file with the specified name. If the file already exists, a message is printed to determine whether the existing file should be superseded. If not, the file is appended to the existing one. Default for file-spec is OPSER.LOG on DSK:.
:TSILENCE n	Suppresses all output from subjob n (same as the :SILENCE command) but places entries into the log file.
:TTYTST	Tests this terminal by printing all the ASCII characters between octal 40 and 174, inclusive.
:WHAT n,m,p	Prints 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 listing of the JOBSTS bits depending upon the value of :MSGLVL and the time of the next timed AUTO file.
:WHERE devn:	Prints the node number of the device's physical location.

When a subjob number or name is required in a command string (indicated by n, m, p), the subjob can be specified in one of four ways.

1. It can be omitted, in which case the last subjob typed into is used.
2. ALL can be specified, in which case all active subjobs are implied.
3. A decimal number can be specified (from 0 to the OPSER limit), in which case a particular subjob number is indicated.
4. A name which was previously assigned to a particular subjob with the :DEFINE command.

## OPSER Program (Cont.)

### Characteristics

The OPSER program:

Requires LOGIN.

Runs OPSER.

Destroys your core image.

Places your terminal at user level.

### Example

1. The following is an example of an automatic startup file.

```
.TYPE SYS:TTY4.ATO (RET)
:TLOG DSKN:OPSER4.LOG
:SLOG
:DEF L1=
:SLOG
:DEF L0=
:SLOG
:DEF M=
:SLOG
:DEF B=
:TSILEN ALL
:L1-AS LPT1
:L1-R LPTSPL
:L1-FREEZ
:L1-MLIMIT 5000
.
```

## PJOB Command

### Function

The PJOB command prints job information on your terminal. If you do not specify a job number, the information printed is about the job to which you are attached. You can specify the job number about which you need information. The PJOB command prints the following information:

- The job number
- The user name
- The project-programmer number
- The terminal number

You receive an error message if no job is attached to the default or specified job number. If the job is currently detached, the letters DET and the terminal number are printed.

### Format

PJOB job

Where: **job** is the number of the job, which is assigned when you log in. If you do not specify the job number, the information printed is about the job you are currently logged in to.

### Characteristics

The PJOB command:

Leaves your terminal at monitor level.

Does not destroy your core image.

### Example

Show information about your job.

```
.PJOB RET
```

```
JOB 42    USER SMITH    [27,4072]    TTY307
```



## PLEASE Command

### Function

The PLEASE command allows you to communicate with the system operator.

### Format

PLEASE dev:prog! text

Where: **dev:** is any terminal not assigned to a job with which you want to communicate. This may be any of the following:

TTYn: where n is the number of any terminal that is not attached to a job. The default is TTY0:

OPRnn: is the operator's terminal at the node number nn.

If you omit this argument, the message goes to TTY0:.

**prog!** is the name of a system program to be run automatically when the message is completed. This argument can appear before or after the device argument and must be ended by an exclamation point. This argument is optional, and cannot be issued from a batch job.

**text** is your message. This argument is required.

When you press the RETURN key, a message informing the operator of your node number, project-programmer number, and text message is printed on the specified terminal, or on the operator's terminal. You and the operator can then type messages without repeating the command. To end communication, press ESCAPE or CTRL/C.

### Characteristics

The PLEASE command:

Places your terminal at user level until you press ESCAPE.

### Restrictions

For batch users, only PLEASE text is allowed. The line of text can only be one line and should be terminated with an ESCAPE.

### Example

Ask the operator when DTA3: will be available. The operator responds by saying "IN HALF AN HOUR." You thank the operator and use CTRL/C to exit from PLEASE.

```
.PLEASE TELL ME WHEN DTA3 WILL BE FREE (RET)
OPERATOR HAS BEEN NOTIFIED IN HALF AN HOUR
THANKS (RET)
```

^C

^C

.

## PLOT Command

### Function

The PLOT command places entries in the plotter output queue. This command is equivalent to the following form of the QUEUE command:

```
QUEUE PLT:jobname=file-spec/switches,file-spec/switches,...
```

### Format

```
PLOT jobname=file-spec/switches,file-spec/switches,...
```

Where: **jobname** is the name of the job you are entering into the queue. The default is the name of the first file in the request.

**file-spec** is the specification of the file or files you want plotted. File specifications are described in Section 1.8. You can use the wildcard constructions in the file specifications.

**/switches** are the switches to the command. The switches are listed in the description of the QUEUE command.

You can obtain a list of the entries in the plotter queue for specific project-programmer numbers by following the command with the project-programmer numbers enclosed in square brackets. If you type PLOT with no arguments, all entries in the plotter queue are printed.

### Characteristics

The PLOT command:

Leaves your terminal at monitor level.

Destroys your core image.

Does not require LOGIN when only a list of the queue entries is desired.

### Example

Plot all files with the extension .PLT. Because they have the default extension .PLT, the files will be deleted after they are plotted. Ask the operator to put PLAIN paper in the plotter.

```
.PLOT *.PLT/FORMS:PLAIN RET
[PLT:GR5=/SEQ:403/LIMIT:10/PRIO:10/FORMS:PLAIN]
```

SYSTEM COMMANDS AND PROGRAMS

PLOT Command (Cont.)

Note that the job name defaults to the first file found with the extension .PLT. Show a list of the jobs in the plotter queue.

.PLOT (RET)

OUTPUT QUEUES:

DEV	JOB	SEQ	PRIO	LIMIT	USER
---	---	---	---	---	---
PLT:	GR5	403	10	10	MAROTTA
/FORMS:PLAIN					

Kill the request:

.PLOT GR5=/KILL (RET)  
[ONE JOB KILLED]

**PRESERVE Command****Function**

The PRESERVE command changes the protection codes of the specified files to <155> or <157>. If the system default is <055>, PRESERVE changes the code to <155>. If the system default is <057>, PRESERVE changes the code to <157>. Protection codes are described in Section 1.8.4.

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

**Format**

PRESERVE file.ext,file.ext,...

Where: **file.ext** is the file name and extension of the file or files to be PRESERVED.

You can use the full wildcard construction for either the file name or the extension, or both.

**Associated Command**

PROTECT - This command allows you to specify the protection code for each file or all files.

**Restrictions**

The PRESERVE command has no effect on sub-file directories (that is, files with the extension .SFD).

**Characteristics**

The PRESERVE command:

Leaves your terminal at monitor level.

Destroys your core image.

**Example**

A DIRECTORY shows that TEST.REL is protected with <055>:

```
.DIR TEST.* (RET)
TEST   FOR    1  <055>   10-SEP-79   DSKC:  [27,5434]
TEST   SFD    1  <755>   28-JAN-80
TEST   REL    1  <055>   31-JAN-80
TEST   MAC    1  <055>    5-FEB-80
TOTAL OF 4 BLOCKS IN 4 FILES ON DSKC: [27,5434]
```

Use PRESERVE to change the protection code of TEST.REL.

```
.PRESERVE TEST.REL (RET)
FILES RENAMED:
DSKC:TEST.REL
```

SYSTEM COMMANDS AND PROGRAMS

PRESERVE Command (Cont.)

Show that the protection of TEST.REL is now <155>:

.DIR TEST.\* RET

TEST	FOR	1	<055>	10-SEP-79	DSKC:	[27,5434]
TEST	SFD	1	<775>	28-JAN-80		
TEST	REL	1	<155>	31-JAN-80		
TEST	MAC	1	<055>	5-FEB-80		
TOTAL OF 4 BLOCKS IN 4 FILES ON DSKC:						[27,5434]

## PRINT Command

### Function

The PRINT command places entries into the line printer output queue. This command is equivalent to the following form of the QUEUE command:

```
QUEUE LPT:jobname=file-spec/switches,file-spec/switches...
```

### Format

```
PRINT jobname=file-spec/switches,file-spec/switches...
```

Where: **jobname** is the name of the job being entered into the queue. The default is the name of the first file in the request.

**file-spec** is the list of file specifications of the file or files to be printed. File specifications are discussed in Section 1.8. You can use wildcard constructions in the file specification.

**/switches** are the switches to the command. The switches are listed in the description of the QUEUE command.

To see a list of all the entries in the print queue, type PRINT with no arguments. To see the entries submitted under a specific PPN, type PRINT followed by the PPN in brackets. The list is printed on your terminal.

### Characteristics

The PRINT command:

Leaves your terminal at monitor level.

Destroys your core image.

Does not require LOGIN when only a list of queue entries is desired.

SYSTEM COMMANDS AND PROGRAMS

PRINT Command (Cont.)

Example

Print the file LOADER.SAVE, which is binary, suppressing all control characters except RETURN and line feed.

```
.PRINT LOADER.SAV/OKBINARY/PRINT:SUPPRESS   
[LPT:LOADER=/SEQ:602/LIMIT:200/OKBINA/PRINT:SUPPRE]
```

Show the list of entries in the line printer queue:

```
.PRINT 
```

OUTPUT QUEUES:

DEV	JOB	SEQ	PRIO	LIMIT	USER
---	---	---	----	-----	-----
LPT02:	HOTT	45	10	550	BILLING
LPT	LOADER	602	10	200	MAROTTA
/OKBINA /PRINT:SUPPRE					

Kill the line printer request:

```
.PRINT LOADER=/KILL   
[ONE JOB KILLED]
```

## PROTECT Command

### Function

The PROTECT command changes the access protection codes associated with the specified files or directories. The change is made by renaming the file.

You specify the access protection of a file with 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 as the owner, and the third represents all other users. Refer to Section 1.8.4 for a description of protection codes.

The standard protection is either <057> or <055>. However, the system default can be changed by individual installations. The owner of a file can alter the protection code of that file regardless of the existing protection code.

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

### Format

```
PROTECT file-spec<nnn>,file-spec<nnn>,...
```

Where: **file-spec** is a file specification. File specifications are described in Section 1.8.

<nnn> is an octal code.

You can specify the protection code before the file name, in which case it applies for all following files. You can use the full wildcard construction for the file specification.

If you have the required privileges, you can change the protection of files not in your directory by specifying the desired directory name. If you specify the directory name before the file name, it applies to all succeeding files specified in the command line.

### Associated Commands

**PRESERVE** - This command changes the protection codes of the specified files to the standard protection code.

### Characteristics

The PROTECT command:

Leaves your terminal at monitor level.

Destroys your core image.



SYSTEM COMMANDS AND PROGRAMS

PROTECT Command (Cont.)

Example

DIRECTORY shows that TEST.REL has the protection <155>:

```
.DIR TEST.*(RET)
TEST   FOR    1  <055>   10-SEP-79   DSKC:   [27,5434]
TEST   SFD    1  <775>   28-JAN-80
TEST   REL    1  <155>   31-JAN-80
TEST   MAC    1  <055>    5-FEB-80
      TOTAL OF 4 BLOCKS IN 4 FILES ON DSKC: [27,5434]
```

Use PROTECT to change the protection code of TEST.REL to <555>:

```
.PROTECT TEST.REL<555>(RET)
FILES RENAMED:
DSKC:TEST.REL
```

Show the new protection of TEST.REL with DIRECTORY.

```
.DIR TEST.*(RET)
TEST   FOR    1  <055>   10-SEP-79   DSKC:   [27,5434]
TEST   SFD    1  <775>   28-JAN-80
TEST   REL    1  <555>   31-JAN-80
TEST   MAC    1  <055>    5-FEB-80
      TOTAL OF 4 BLOCKS IN 4 FILES ON DSKC: [27,5434]
```

## PUNCH Command

### Function

The PUNCH command duplicates the CPUNCH command at most installations. However, it is possible for the installation to change this function so that PUNCH duplicates TPUNCH.

## QUEUE Command

### Function

The QUEUE command allows you to make entries in several system queues - the input queue for the batch system and the output spooling queues for the line printer, card punch, paper tape punch, and plotter. You can also use the QUEUE command to print a list of the queue entries on your terminal.

The following commands run the QUEUE program. Each command accesses a system queue. The QUEUE command accesses all the system queues. Each command and the queue that it accesses is listed below:

Command	Queue
CPUNCH	Card punch
PLOT	Plotter
PRINT	Line printer
SUBMIT	Batch input
TPUNCH	Paper tape punch

### Formats

QUEUE INP:jobname=control-file-spec,log-file-spec/switches

Makes an entry in the batch input queue.

QUEUE queue-name:jobname=file-spec/switches

Makes an entry in an output queue.

QUEUE file-spec/LIST=queue-name.

Creates a file containing a list of queue entries. If you omit the file-spec, the switch, and the queue-name, all the entries in all the queues are printed on your terminal. The list of entries includes the job name, sequence number, job state, job priority, limit, and user name.

Where: **control-file-spec** 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 you have access to; the default is DSK:. The file name is required, but you can omit the extension; the default is .CTL. The wildcard constructions are legal for the file name or extension and is the only way you can submit multiple control files to batch with one QUEUE command.

**log-file-spec** is the file specification for the file that is to be used to record the execution of the control file. This file can be written in any directory on any file structure that you have this kind of access to. The default is the same file structure in which the control file resides. If the file name is missing, the log file is given the same name as the control file. If the extension is omitted, the default is .LOG.

**QUEUE Command (Cont.)**

**jobname** is the name of the job being entered into the queue. The default job name is the name of the first file in the request. This may differ from the first file specified in the string, if the first file does not yet exist, if the wildcard construction is used, or if an indirect command file is used.

**file-spec** is the list of file specifications for the files to be output. The files can be on any file structure that your job has access to; the default is DSK:.. The files can be in any directory, provided that you have access to them; the default is your directory. The file name is optional; the default is \* for the first file name. The default for subsequent file names is the last file name used. You can omit the file extension because the file has the default extension. These default extensions are:

Extension	Queue
.CTL	Batch input queue
.LPT	Line printer queue
.CDP	Card punch queue
.PTP	Paper-tape punch queue
.PLT	Plotter queue

**queue-name** is the name of the queue to be accessed. If output, the queue-name is the kind of queue. If input, it is the batch processing queue. The queue-names follow:

Queue	Queue-name
Batch	INP:
Line printer	LPT: (L:)
Card punch	CDP: (C:)
Paper tape punch	PTP: (PT:)
Plotter	PLT: (PL:)

**/switches** are switches that further define the command. The switches affect the function of the queue, the processing of the file, and the kind of queue. Therefore, some switches affect the whole command string, others affect the files, and some affect a single file. The three categories of switches and their functions follow:

1. **Queue operation switches** define the type of request. Therefore, you can type only one of these switches in the string. The switch can be placed before or after the file-spec.
2. **General switches** affect the function of the queue. Therefore, you can type each switch only once in the string. You can place them before or after the file-spec.

**QUEUE Command (Cont.)**

3. **File control switches** affect the files in the request. You can use as many of them as often as necessary. However, their placement in the string defines the files they affect. If you type one of these switches before a file-spec, it becomes the default for all following file-specs, until you type a switch that changes it. If you type the switch after a file-spec and separate it from any following file-specs with a comma, it affects only the file-spec it follows. Refer to the examples.

The switches are also defined by the queue(s) that they affect. The following conventions are used:

Convention	Meaning
ALL	All system queues.
INPUT	Batch queue.
OUTPUT	All output queues.
(xxx)	The specific queue only.
LIST	No specific queue; these affect the lists of queue entries.

If you omit an argument to a switch, you must also omit the colon preceding the argument. Otherwise, the argument is assumed to be zero and not the default value.

Switch	Category	Queues	Function
/AFTER: date-time	General	ALL	Processes the request after the specified time. Refer to Section 1.7.3 for a description of date-time arguments. The result must be less than the DEADLINE time.
/BEFORE: date-time	General	OUTPUT	Queues only the files with creation dates before date-time. The date must be today or earlier. Refer to Section 1.7.3 for a description of date-time arguments.
/BEGIN:n	File control	OUTPUT(LPT) INPUT	Starts the output on the nth page for line printer requests or on the nth line of the control file for batch requests. The default is to begin output on the first unit.

SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/CARDS:n	General	INPUT	Uses n as the maximum number of cards that can be punched by your batch job (up to 10,000). If you omit the switch, no cards will be punched. If you omit the value, the default is 2000.
/COPIES:n	File control	OUTPUT	Repeats the output the specified number of times (n must be less than 64). The default is one copy. If you need more than 63 copies, make two or more requests.
/CORE:n	General	INPUT	Uses n (in decimal K) as the maximum amount of memory that your job can use. If you omit the switch, the maximum is 25K. If you omit the value, the default is 40K.
/CREATE	Queue operation	ALL	Makes a new entry in the specified queue. This switch is the default for queue requests.
/DEADLINE: date-time	General	ALL	Processes the request before the specified date-time. Refer to Section 1.7.3 for a description of date-time arguments. The result must be greater than the /AFTER time.
/DELETE	File control	ALL	Deletes the file after the job is finished. Same as /DISPOSE:DELETE.
/DEPEND:n	General	INPUT	Specifies the initial value of the dependency count (n) 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 job's dependency count is changed to n.
/DISPOSE: DELETE	File control	ALL	Deletes the file after spooling it.
/DISPOSE: PRESERVE	File control	ALL	Saves the file after spooling. This is the default for all files except those with the default output extensions.

SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/DISPOSE: RENAME			Removes the file from the specified directory, removes it from the logged-out quota, and deletes it after spooling it. This is the default for files with the default extensions and with .TMP.
/ERBINARY	General	ALL	Prints an error message if a binary file is included in the queue. This is the default.
/ERNONE	General	ALL	Prints an error message if no files match the wildcard construction. This is the default.
/ERPROTECTION	General	ALL	Prints an error message if processing the request requires a violation of protection codes. This is the default function.
/FAST	Queue operation	LIST	Prints a short list of the entries in the queues.
/FEET:n	General	INPUT	Uses n as the maximum number of feet of paper tape that your job can punch. If you omit the switch, no paper tape is punched. If you omit the value, the default is 10xB+20 feet, where B is the number of blocks in the file.
./FILE:ASCII	File control	OUTPUT	Specifies that the file is to be interpreted as ASCII text. This is the default for all files with extensions other than .DAT.
/FILE:COBOL	File control	OUTPUT (LPT)	Specifies that the file is to be interpreted as COBOL SIXBIT text.
/FILE:ELEVEN	File control	OUTPUT (PTP) (CDP)	Specifies that the file 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 is bits 2-9. 3rd byte is in bits 28-35. 4th byte is in bits 20-27.

SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/FILE:FORTRAN	File control	OUTPUT (LPT)	Specifies that the file is to be interpreted as FORTRAN ASCII text. This is assumed if the file extension is .DAT.
/FORMS:a	General	OUTPUT (LPT)	Prints the file(s) on the specified forms. The argument to the switch must be six alphanumeric characters or less. The default forms are NORMAL. NARROW forms are 8 1/2x11 inches.
/HEADER: 0 or 1	File control	OUTPUT (LPT)	Prints header pages between files if you specify 1. Does not print headers between files if you specify 0.
/HELP or /HELP:TEXT			Prints a message giving the general format of the command string and explains the dialogue you must enter if you need additional help.
/HELP: SWITCHES			Prints a list of QUEUE switches.
/JOBNAME:name	General	ALL	Specifies the name of the job. The name can be up to six alphanumeric characters.
/KILL	Queue operation	ALL	Removes the specified entry from the specified queue. You must enter the job name or sequence number, followed by an equal sign (=), in the command string.
/LIMIT:n	General	OUTPUT	Limits the output to the specified number of pages, cards, feet, or minutes. One page is about three blocks.
/LIST	Queue operation	LIST	Prints a list of the entries in the queue. The default is all entries in all queues.
/L:ALL	Queue operation	LIST	Prints all data about each queue request on your terminal. This is the default.
/L:DETAIL	Queue operation	LIST	Prints detailed job data on each queue request.
/L:FAST	Queue operation	LIST	Prints a short list of queue entries on your terminal.



SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/L:FILES	Queue operation	LIST	Prints a list of all the files in each queue request.
/L:JOBS	Queue operation	LIST	Prints a list of jobs in the queue. This is the default.
/LOCATE:nnn	General	ALL	Specifies the remote station number (nnn) of the job where the output is to be sent. The default function is to output the files at the station where the request was entered.
/LOG:arg	File control	INPUT	Defines the file that the spoolers will record processing. The default file name is jobname.LOG.
/METERS:n or /METRES:n	General	INPUT	Specifies the number of meters of paper tape that can be punched by the job.
/MODIFY	Queue operation	ALL	Alters the specified parameters in the specified jobs. This switch requires that you have access to the job. You must specify the job name or sequence number of the job, followed by an equal sign (=) in the command string.
/NEW	File control	ALL	Accepts the request even if the file does not yet exist. This is the default for the log files of batch input queues.
/NOHEADER	General	OUTPUT (LPT)	Suppresses printing of header pages between files.
/NONEW	File control	ALL	Forces the system to use an existing file, if possible. This is the default for jobs entered from interactive level.
/NONULL	General	OUTPUT	Prints an error message if no files are processed by the request. This is the default for all queue requests.
/NOOPTION	General	OUTPUT	Suppresses the SWITCH.INI file.
/NOPHYSICAL	General	OUTPUT	Allows logical names for devices in the command string. This is the default.

SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/NOREMOVE	File control	OUTPUT	Does not remove a file from the queue. This switch is useful with /KILL, to define files that should remain in the queue.
/NORESTART	General	INPUT	Prevents your job from being restarted when it was stopped by a system crash.
/NOSTRS	File control	OUTPUT	Does not scan each structure for files with the same name. This is the default.
/NOTE:text	General	OUTPUT (LPT)	Prints the text on the header pages. The text can be up to 12 characters.
/NOUNPRE	General	OUTPUT	Allows any file to be output.
/NULL	General	OUTPUT	Does not print a fatal error message if the specified files do not exist.
/OKBINARY	File control	OUTPUT (LPT)	Prints files whose extensions indicate binary information. Normally files with extensions .SAV, .SHR, .LOW, .REL, and .HGH will not be accepted in the print queue.
/OKNONE	File control	OUTPUT	Does not produce a warning message if no files match the wildcard construction.
/OKPROTECTION	General	ALL	Ignores all protection errors.
/OPTION:option	General	ALL	Uses the line QUEUE:option in the SWITCH.INI file. SWITCH.INI files are described in Appendix B.
/OUTPUT:arg	General	INPUT	Determines whether or not the log file will be printed. This switch has three arguments, only one of which can be specified in the string. The arguments are LOG, NOLOG, and ERROR.  LOG Prints the log file.  NOLOG Suppresses printing of the log file.  ERROR Prints the log file only if an error occurs.

SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/PAGE:n	General	INPUT	Uses n as the maximum number of pages that your job can print. If you omit the switch, the default is 200 pages. If you omit the value, the default is 2000.
/PAPER:x	File control	OUTPUT	Identical to /PUNCH:x., /PRINT:x, /TAPE:x, or /PLOT:x.
/PATH:[dir]	File control	INPUT	Runs the job from the specified directory.
/PHYSICAL	File control	ALL	Suppresses logical device names for the specified device.
/PLOT:ASCII	File control	OUTPUT (PLT)	Plots the file in ASCII mode. If you omit this switch, the file is plotted in the data mode specified in the file.
/PLOT:BINARY	File control	OUTPUT (PLT)	Plots the file in binary mode. If you omit this switch, the file is plotted in the data mode specified in the file.
/PLOT:IMAGE	File control	OUTPUT (PLT)	Plots the file in image mode. If you omit this switch, the file is plotted in the data mode specified in the file.
/PRESERVE	File control	All	Saves the file after spooling it. This is the default for all files except those with the .LST extension. This switch is the same as /DISPOSE:PRESERVE.
/PRINT:ARROW	File control	OUTPUT (LPT)	Converts all control characters to up-arrow format except characters 011 through 015 and 020 through 024.
/PRINT:ASCII	File control	OUTPUT (LPT)	Sends the file to the line printer with no changes. This is the default.
/PRINT:OCTAL	File control	OUTPUT (LPT)	Prints the file in octal.
/PRINT: SUPPRESS	File Control	OUTPUT (LPT)	Converts all carriage-control characters to line feeds except for ASCII code characters, CR and DC3. This switch implies /PRINT:ARROW.

SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/PRIORITY:n	General	ALL	Gives the specified priority (n is 1 to 63) to the request. A larger number has greater priority. If you omit the switch, the default value is 10. If you omit the value, the default is 20.
/PUNCH:026	File control	OUTPUT (CDP)	Punches files in 026 Hollerith code. If you omit this switch, the files are punched according to the data mode specified in the file.
/PUNCH:ASCII	File control	OUTPUT (CDP)	Punches files in ASCII card code. If you omit this switch, the files are punched according to the data mode specified in the file.
/PUNCH:BCD	File control	OUTPUT (CDP)	Punches the files in 026 Hollerith code. If you omit this switch, the files are punched according to the data mode specified in the file. Same as /PUNCH:026.
/PUNCH: BINARY	File control	OUTPUT (CDP)	Punches files in checksummed binary card format. If you omit this switch, the files are punched according to the data mode specified in the file.
/PUNCH:IMAGE	File control	OUTPUT (CDP)	Punches files in image card format. If you omit this switch, the files are punched according to the data mode specified in the file.
/PUNCH: MNEMONIC	File control	OUTPUT (CDP)	Punches files in mnemonic binary mode. If you omit this switch, the files are punched according to the data mode specified in the file.
/REMOVE	File control	ALL	Removes 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 your job.

SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/REPORT:code	File control	OUTPUT (LPT)	Prints the specified report cover page and up to one additional page within a COBOL report file. The code can be up to 12 characters in length.
/RESTART: 0 or 1	General	INPUT	Specifies whether job can be restarted by the operator after a system crash. If you specify 0 (or NO), the job cannot be restarted. If you specify 1 (or YES), the job can be restarted. Do not specify 1 (or YES) if the job makes changes to your directory area.
/RUN:file	File control	OUTPUT	Executes the specified file after the job is done.
/RUNCORE:n	File control	OUTPUT	Executes the specified file in nK of core after the job is done.
/RUNOFFSET:n	File control	OUTPUT	Executes the specified file with offset n after the job is done.
/SEQUENCE:n	General	ALL	Specifies a sequence number to aid in identifying a request to be modified or deleted.
/SINCE: date-time	General	OUTPUT	Queues only the files with creation dates after the specified date and time, in the form dd-mmm-yy:hh:mm.
/SPACING: DOUBLE	File control	OUTPUT (LPT)	Creates double-spaced output.
/SPACING: SINGLE	File control	OUTPUT (LPT)	Creates single-spaced output. This is the default.
/SPACING: TRIPLE	File control	OUTPUT (LPT)	Creates triple-spaced output.
/STRS	File control	OUTPUT	Searches for the file on all structures in the search list and takes each occurrence. The default is to take just the first occurrence of the file.
/TAG:xxx	File control	INPUT	Starts at the statement labeled xxx in the control file. Equivalent to GOTO xxx at the beginning of the control file.

SYSTEM COMMANDS AND PROGRAMS

QUEUE Command (Cont.)

Switch	Category	Queues	Function
/TAPE:ASCII	File control	OUTPUT (PTP)	Punches the tape in ASCII code. If you omit this switch, the file is punched according to the mode specified in the file.
/TAPE:BINARY	File control	OUTPUT (PTP)	Punches the tape in binary mode. If you omit this switch, the file is punched according to the mode specified in the file.
/TAPE:IBINARY	File control	OUTPUT (PTP)	Punches the tape in image-binary mode. If you omit this switch, the file is punched according to the mode specified in the file.
/TAPE:IMAGE	File control	OUTPUT (PTP)	Punches the tape in image mode. If you omit this switch, the file is punched according to the mode specified in the file.
/TIME: hh:mm:ss	General	INPUT	Specifies the CPU time limit for your job. If you omit this switch, the default time is 5 minutes. If you omit the value, the default is one hour.
/TPLOT:n	General	INPUT	Uses n (decimal minutes) as the maximum amount of plotting time allowed for your job. If you omit the switch, no plotter time is allowed. If you omit the value, the maximum plotter time is 10 minutes.
/UNIQUE: 0 or 1	General	INPUT	Specifies job capability. If the value is 0 (or NO), any number of batch jobs can run from this PPN at one time. If the value is 1 (or YES), only one batch job can run from this PPN at a time.
/UNPRESERVED	General	OUTPUT	Processes the file only if it is not preserved.

**QUEUE Command (Cont.)****Characteristics**

The QUEUE command:

Leaves your terminal at monitor level.

Destroys your core image.

Does not require LOGIN when only a list of queue entries is desired.

**Example**

Queue the files ONE.LPT and LOGOUT.TRN. They are automatically queued to the line printer. ONE.LPT will be deleted after it is printed because it has the default extension .LPT.

```
.QUEUE ONE.LPT,LOGOUT.TRN(RET)
[LPT:=ONE/SEQ:23/LIMIT:430/PRIO:10
```

Queue the file BACKP.CTL to the batch queue:

```
.QUEUE INP:=BACKP(RET)
[INP02:BACKP=/SEQ:301/TIME:00:05:00/CORE:20P]
```

Show all the requests in all the queues:

```
.QUEUE(RET)
```

## INPUT QUEUE:

STS	JOB	SEQ	PRIO	TIME	CORE	USER
---	---	---	----	-----	-----	-----
RUN	BACKP	301	10	00:05:00	20P	MAROTTA

## OUTPUT QUEUES:

DEV	JOB	SEQ	PRIO	LIMIT	USER
---	---	---	----	-----	-----
LPT0:	ONE	23	10	430	MAROTTA

.

## QUOLST Program

### Function

The QUOLST program prints on your terminal the amount of disk space you have used and the amount you have left on each file structure in your search list. In addition, it prints the amount of space the file structures have left for all users.

For each file structure, the following information is given:

- The structure name.
- The number of blocks allocated.
- The number of blocks left in the logged-in quota, in the logged-out quota, and on the structure.

### Format

R QUOLST

### Characteristics

The QUOLST program:

Destroys your core image.

Leaves your terminal at monitor level.

### Example

.R QUOLST RET

USER:	27,5434				
STR	USED	LEFT: (IN)	(OUT)	(SYS)	
DSKC:	1355	8645	3645	61995	
DSKB:	0	10000	5000	96870	



## R Command

### Function

The R command loads a core image from the system device (SYS:) and starts it at the location specified within the file (.JBSA in the Job Data Area. See the TOPS-10 Monitor Calls Manual). This command is the same as RUN SYS:file.ext. The R command is used to run a system program that does not have an operating system command to run it.

This command clears all user core. However, programs should not count on this action and should explicitly clear those areas of core that are expected to contain zeros (that is, programs should be self-initializing). This action allows programs to be restarted by a CTRL/C, START sequence without another R command.

### Format

R file.EXE[directory] core

Where: **file.EXE** is the file name of a file stored on the SYS: file structure.

### NOTE

Old-style SAVE files (.SAV, .HGH, .SHR) can be run by the R command. However, .EXE files are recommended over SAVE files.

[directory] is a directory specification. If a directory specification is present in the command string, it overrides the assumed SYS: device. Directory paths are described in Section 1.13.

core is the amount of core needed to run the program. If you do not specify core, the default is the minimum amount of core needed to load the program into memory.

If you do not specify an argument, the program currently in memory is run.

### Characteristics

The R command:

Destroys your core image.

## R Command (Cont.)

### Example

Run the SYS:SETSRC.EXE program

```
.R SETSRC 
```

SETSRC prompt.

```
*^C
```

Use CTRL/C to exit.

.

## REASSIGN Command

### Function

The REASSIGN command passes a device assignment from one job to another job. Both restricted and unrestricted devices can be reassigned. When you specify a DECTape in this command, the system clears the copy of the directory currently in memory. The next time you refer to the directory, a new copy is read from tape into memory. However, the logical name that has been assigned is not reassigned unless you reassign the device to your own job.

### Format

REASSIGN dev:job

Where: **dev:** is the physical or logical name of the device to be reassigned. This argument is required. Refer to Section 1.8.1 for a description of device names.

**job** is the number of the job to which the device is to be reassigned. If no job is specified, the device is reassigned to your job. This is useful when you want to force the next DECTape directory reference to come from the tape instead of core.

You can reassign a device using the logical name only if your job and the job to which the device is to be reassigned have the same project-programmer number, or you have operator privileges (logged in under [1,2] or logged in as OPR).

### Associated Messages

If the job number you specify is out of the range of legal job numbers, the monitor prints:

?NOT A JOB

If no arguments are included in the REASSIGN command, the monitor prints:

?NOT ENOUGH ARGS

If the device specified is not assigned to your job, the monitor prints:

?CAN'T BE REASSIGNED

### Characteristics

The REASSIGN command:

Leaves your terminal at monitor level.

Will not function during device I/O.

### Restrictions

Your job's controlling terminal cannot be reassigned.

SYSTEM COMMANDS AND PROGRAMS

REASSIGN Command (Cont.)

Example

Assign a device to a job.

```
.ASSIGN CDR: (RET)  
CDR261 ASSIGNED
```

A SYSTAT of your PPN shows you are running two jobs.

```
.SYS [,] (RET)  
  
13      31      SYSTAT  19+SPY  RN      12$  
19      DET      SETSRC  30+4    ^C SW  
$ MEANS EXECUTE ONLY
```

```
BUSY DEVICES:  
DEVICE  JOB      WHY      LOGICAL  
CDR261  13      AS
```

The card reader is assigned to job 13.

Reassign the device to job 19.

```
.REASSIGN CDR:19 (RET)
```

SYSTAT shows that the reassignment was successful.

```
.SYS [,] (RET)  
  
13      31      SYSTAT  24+SPY  RN      12$  
19      DET      SETSRC  30+4    ^C SW  1  
$ MEANS EXECUTE ONLY
```

```
BUSY DEVICES:  
DEVICE  JOB      WHY      LOGICAL  
CDR261  19      AS
```

**REATTACH Command****Function**

The REATTACH program transfers your job from one terminal to another. Unlike the ATTACH command, REATTACH does not require a password or that the terminal be of the same type that LOGIN recognizes to run the job. For example, usually a [1,2] job can run only on a local terminal. However, the REATTACH program can be used to reattach a [1,2] job from a local terminal to a remote terminal.

Before reattaching your job, verify that the terminal to which you are attaching is turned on and working properly. Otherwise, it might be difficult to retrieve the job.

**Format**

R REATTACH

REATTACH responds by asking for the new terminal name.

NEW TTY NAME:

You answer by typing either the new terminal name (for example, CTY, TTY2) or number (for example, 2). REATTACH then responds with

FROM JOB n

on the old terminal, and

NOW ATTACHED TO JOB n

on the new terminal.

**Characteristics**

The REATTACH program:

Leaves your terminal at monitor level.

Destroys your core image.

**Restrictions**

The REATTACH program is not available to batch jobs.

**Example**

The following appears on the old terminal.

```
.R REATTA RET
TYPE NEW TTY NAME:  TTY27
FROM JOB 7
```

.

The following appears on TTY27.

```
NOW ATTACHED TO JOB 7
```

.

**REENTER Command****Function**

The REENTER command restarts a program at an alternate entry point. You must write the alternate entry point into your program. The REENTER command is used for a partial reinitialization of the program.

The REENTER command copies the saved program counter value from .JBPC in the Job Data Area into .JBOPC and starts the program at an alternate entry point specified in .JBREN (you must set it or have it set in your program). All Job Data Area locations are described in the TOPS-10 Monitor Calls Manual.

If the job was executing a monitor call when it was interrupted (that is, at monitor level but not in TTY input wait or SLEEP mode), the monitor continues the job until the monitor call 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 monitor call.

If the job is at user level, the trap occurs immediately. Therefore, it is always possible to continue the interrupted program after trapping by executing a JRSTF @.JBOPC.

**Format**

```
REENTER
```

**Associated Commands**

The START command is used for complete reinitialization of a program.

The CONTINUE command is used when you want to continue the program without reinitialization.

**Characteristics**

The REENTER command:

- Places your terminal at user level.

**Example**

Run the SYSTAT program for job 35:

```
.SYS 35 (RET)
```

```
35 20,641 TTY45 SOS 12+24 T1 7#
#MEANS NON-SYSTEM HI-SEG
```

Reenter the program:

```
.REENTER (RET)
```

```
35 20,641 TTY45 SOS 12+24 T1 0 2#
TOTAL KCS USED = 2 AVERAGE KCS=2
#MEANS NON-SYSTEM HI-SEG
```

**RENAME Command****Function**

The RENAME command changes one or more attributes of the file specification of files on disk or DECTape. Refer to Section 1.8 for a description of file specifications.

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

**Format**

```
RENAME file-spec=file-spec,file-spec=file-spec,...
```

Where: the new file-spec is placed first, to the left of the equal sign (=).

the old file-spec is placed at the right of the equal sign (=).

Device and file-structure names cannot be different for old and new file-specs. If you specify the device name in the new file-spec, that device is the default for the old file-spec. You can use RENAME to change the protection code of a file by specifying the protection codes in the file-specs.

The directory name remains effective throughout the command line, but you can specify a different directory name to change the directory of a file. RENAME accepts wildcard constructions.

If the new file name is specified without an extension, the file will not have an extension.

**Characteristics**

The RENAME command:

Leaves your terminal at monitor level.

Destroys your core image.

Requires LOGIN.

**Example**

Show the file named PROG.

```
.DIR PROG.*RET
PROG   EXE   68   <055>  10-FEB-80   DSKC:[27,5434]
```

Rename PROG to be PRGRM.

```
.RENAME PRGRM.EXE=PROG.EXERET
FILES RENAMED:
DSKC:PROG.EXE
```

SYSTEM COMMANDS AND PROGRAMS

RENAME Command (Cont.)

Show the file named PRGRM.

.DIR PRGRM.\*

PRGRM EXE 68 <055> 10-FEB-80 DSKC:[27,5434]



## RESOURCES Command

### Function

The RESOURCES command prints the names of all available devices (except terminals and pseudo-terminals), all file structures, and all physical units not in file structures.

### Format

RESOURCES

### Characteristics

The RESOURCES command:

Leaves your terminal at monitor level.

Does not require LOGIN.

### Example

```
.RESOURCES(RET)
OAK,BSIC,CCBL,ZINA,DSKA,DSKB,DSKC,DSKN,FHA1,RPA1,RPA4,RPA5,DPA3,DPA4,DPA
6,DPB0,DPB1,CDR010,011,PTR010,LPT011,DTA011,012,014,015,DTB010,011,012,M
TA010,011,012,013,014,015,MIC010,012,013,PTP010,CDP012,PLT010,011,DIS010
,PAR010,011,PCR010,PAP010,011,LPC010,PCP010
```

## REWIND Command

### Function

The REWIND command rewinds a magnetic tape or a DECTape.

### Format

REWIND dev:

Where: **dev:** is a magnetic tape (MTxnnu) or a DECTape (DTxnnu).

Refer to Section 1.8.1 for a description of device names.

### Characteristics

The REWIND command:

Leaves your terminal at monitor level.

Destroys your core image.

Runs the PIP program.

### Examples

1. Rewind DECTape unit 4:

```
.REWIND DTA4:RET
```

.

2. Rewind magnetic tape unit 1:

```
.REWIND MTAl:RET
```

.

## RUN Command

### Function

The RUN command starts execution of a program. It loads a core image from a retrievable storage device and starts the program at the location specified within the file (.JBSA).

RUN looks for programs stored in files with the extension .EXE. Use SAVE or SSAVE to create the executable file.

The RUN command clears all your core. However, programs should not count on this action and should explicitly clear those areas of core that are expected to contain zeros (that is, the programs should be self-initializing). This allows programs to be restarted by a CTRL/C, START sequence without having to issue another RUN command.

### Format

RUN dev:file.EXE[directory] core

Where: **dev:** is the logical or physical name of the device containing the core image. The default device name is DSK:.

**file.EXE** is the name of the file that contains the program you want to run.

### NOTE

Old-style SAVE files (.SAV, .HGH., .SHR) can be run with the RUN command. However, the .EXE format is recommended over the .SAV format.

[**directory**] is the directory name, which is required only if the core image file is located in a disk area other than yours. If a directory specification is issued in the command string and the device is SYS:, the specified directory overrides the assumed device (SYS:).

**core** is the amount of core if different from the minimum core needed to load the program or if it is different from the core argument of the SAVE command that saved the file.

If core is greater than 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 is greater than 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.

## SYSTEM COMMANDS AND PROGRAMS

### RUN Command (Cont.)

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.

#### Characteristics

The RUN command:

Places your terminal at user level.

#### Example

Type a test program.

```
.TYPE PROG.FOR(RET)
TYPE 69
69  FORMAT ('TESTING EXECUTION')
END
```

Load the program.

```
.LOAD PROG(RET)
LINK: LOADING
[LNKCXT PROG EXECUTION]
```

```
END OF EXECUTION
CPU TIME: 0.02 ELAPSED TIME: 0.12
EXIT
```

Save the program, creating an executable file.

```
.SAVE(RET)
PROG SAVED
```

Run the program.

```
.RUN PROG(RET)

TESTING EXECUTION

END OF EXECUTION
CPU TIME: 0.02 ELAPSED TIME: 0.07
EXIT
```

## SAVE Command SSAVE Command

### Function

The SAVE and SSAVE commands create executable files from programs that are loaded in core memory. They create a core image file from a source file. SAVE creates non-sharable files with the extension .EXE. SSAVE creates sharable files with the extension .EXE.

After you use SAVE or SSAVE to create an .EXE file, you can run the program with the RUN command.

### NOTE

Some monitor installations create core image files that are segmented. SAVE and SSAVE will create segmented core image files at such an installation. Use NSAVE and NSSAVE to specify that the core image be saved in an .EXE file, which is not segmented. .EXE files are preferred over old-style files.

### Format

SAVE dev:file.EXE[directory] core

Where: **dev:** is the device on which the core image file is to be written. The default device name is DSK:. The colon following the device name is required if a device is specified.

**file** is the name to be assigned to the core image file. The default file name is your job's current name as set by your last R, RUN, or GET command; the last command that ran a program (for example, DIRECT); or the last SETNAM monitor call.

**.EXE** is the default and preferred file extension of the core image file. This argument is optional.

**[directory]** is the location on the disk area where the core image file is to be written.

**core** is 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.

You can specify core arguments 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 you do not specify K or P, K (1024 words) is assumed.

If core is omitted, only the number of blocks required by the core image area (as explained in the RUN command description) is assumed.

**SAVE Command (Cont.)****Characteristics**

The SAVE command:

- Leaves your terminal at monitor level.
- Does not operate while I/O is in progress.
- Requires memory.

**Example**

Load a program.

```
.LOAD PROG(RET)
FORTRAN: PROG
MAIN.
LINK: LOADING
EXIT
```

Save the file with SAVE.

```
.SAVE PROG(RET)
PROG SAVED
```

Use DIRECTORY to see the new .EXE file.

```
.DIR PROG(RET)

PROG      FOR      1 <055>    10-SEP-79    DSKC: [27,5434]
PROG      REL      1 <055>    10-SEP-79
PROG      EXE     68 <055>    10-SEP-79
TOTAL OF 131 BLOCKS IN 5 FILES ON DSKC: [27,5434]
```

Run the program.

```
.RUN PROG(RET)

THIS IS PROGRAM OUTPUT

END OF EXECUTION
CPU TIME: 0.02  ELAPSED TIME: 0.07
EXIT
```

**SCHED Command****Function**

The SCHED command prints the schedule bits set by the operator. You can obtain this information before you log in to determine the use of the system (for example, regular timesharing or batch jobs only). The schedule bits are as follows:

Bits	Meaning
0000	Regular timesharing.
0001	No further LOGINS allowed except from CTY.
0002	No further LOGINS from remote terminals, and no answering of data sets.
0004	Batch jobs only.
0100	Device mounts can be done without operator intervention.
0200	Unspooling allowed.
0400	No operator coverage.
1000	No automatic downline loading of nodes.

**Format**

SCHED

**Characteristics**

The SCHED command:

Leaves your terminal at monitor level.

Does not require that you be logged in.

**Examples**

1. Regular timesharing.

```
.SCHED (RET)
000000
```

.

2. No operator coverage.

```
.SCHED (RET)
000400
```

.

SYSTEM COMMANDS AND PROGRAMS

**SCHED Command (Cont.)**

3. No LOGINS allowed from local or remote terminals and data sets are not answered.

```
.SCHED (RET)  
000002
```

.

4. Regular timesharing, but no operator coverage. Device mounts can be accomplished without operator intervention.

```
.SCHED (RET)  
000500
```

.



**SEND Command****Function**

The SEND command provides one-way interterminal communication. A single line of information is transmitted from the initiating terminal to the terminal specified in the SEND command string. The recipient of the SEND receives the terminal line number of the sender as well as the message.

A busy test is made on a single-destination message before the message is sent, unless the sender or receiver of the message is OPR or a job logged in as [1,2]. The receiver of the message is considered busy if the terminal is not at monitor command level. If the receiving terminal is busy, the sender receives the message BUSY and the information is not sent. However, the receiving terminal always receives the message if it has the TTY NO GAG bit set (refer to the SET TTY NO GAG command).

If the receiving terminal is hardwired and turned off, the information appears to have been sent because the software cannot detect an OFF condition on hardwired terminals. Terminals in IMAGE mode do not receive messages.

**Formats**

SEND TTYn: text

SEND JOB n text

Where: **TTYn:** is any physical terminal name, including CTY: (console terminal) and OPR:. If you specify OPR:, the system sends the message to the operator's terminal at the node to which your terminal is connected.

n is the job-number to which the message is to be sent.

**text** is the message to be transmitted.

The message is printed on the receiving terminal in the following format:

```
;;TTYn: - text
```

where n is the sender's terminal number and text is the message. A bell sounds on the receiving terminal when the message is printed.

You can specify an operator on a particular node by entering the terminal identifier in the form:

OPRnn:

where nn is the number of the operator's node. See Example 4 below.

## SEND Command (Cont.)

### Characteristics

The SEND command:

Leaves your terminal at monitor level.

Does not require LOGIN.

### Restrictions

The SEND command is not available to batch jobs.

### Examples

1. Send a message to TTY62:

```
.SEND TTY62: YOU HAVE THE SYSTEM STAND-ALONE (RET)
```

.

2. Ask your operator to write-enable a DECTape drive:

```
.SEND OPR: PLEASE WRITE-ENABLE DTA3: (RET)
```

.

3. Send a message to job 69:

```
.SEND JOB 69 THE TAPE IS WRITE ENABLED (RET)
```

.

4. Send a message to the operator on node 17:

```
.SEND OPR17: LPT171 UPPER CASE ONLY? (RET)
```

.

## SET BLOCKSIZE Command

### Function

The SET BLOCKSIZE command sets a default blocksize (in words) for the specified magnetic tape.

### Format

```
SET BLOCKSIZE dev:nnnn
```

Where: **dev:** is either the physical device name, written in the format MTXnnn:, or the logical name previously associated with the device. The device must be assigned to your job. This argument is necessary. See Section 1.8.1 for information about device names.

**nnnn** is 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 in addition to the check for the maximum 4094 and a minimum of 3. This argument is required.

### Characteristics

The SET BLOCKSIZE command:

Leaves your terminal at monitor level.

### Example

Assign the logical name WEEK2 to magnetic tape unit 4.

```
.ASSIGN MTA4:WEEK2 (RET)  
MTA4 ASSIGNED
```

Set the block size of magnetic tape unit 4 to 2000.

```
.SET BLOCKSIZE WEEK2: 2000 (RET)
```

**SET BREAK Command****Function**

The SET BREAK command is used during the debugging process (K110 and K110 processors only). It is useful when the program that 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 your address space. This is useful when the monitor stores or retrieves arguments to/from unexpected locations in your address space because of erroneous monitor call argument lists.

If you are breaking on a WRITE condition, the WRITE condition causing the break will not have been executed. Therefore, the instruction located at PC and all operands should be examined before continuing program execution.

**Formats**

SET BREAK

SET BREAK AT addr ON condition AFTER n,...

SET BREAK NO condition

SET BREAK NONE

Where: **AT**, **ON**, and **AFTER** are optional portions of the command line.

**addr** is an octal number in the range 0-777777, representing a user virtual address. The address is not necessary.

**condition** is one or more reasons for allowing the break to occur. The condition is not necessary. The condition arguments are listed below.

**n** is the number of times the sequence should run before the break occurs. The variable is any decimal number from 1 to 511.

Multiple conditions can be specified within one command; these conditions are separated from one another by commas. The possible conditions that can be specified are:

## SYSTEM COMMANDS AND PROGRAMS

### SET BREAK Command (Cont.)

Condition	Function
READ	Breaks if the contents of addr are read by the program. This condition causes a break to occur on a read-modify-write as well as on a read.
WRITE	Breaks if the location specified by addr is written into.
EXECUTE	Breaks if an instruction is fetched from the location specified in the command string.
ALL	Breaks if the location specified in the command string is read from (READ), written into (WRITE), or fetched from (EXECUTE).
MUO	Breaks on monitor references as qualified by READ, WRITE, and/or EXECUTE.

If all three break conditions are to be specified, you can include the word ALL in the command line, replacing the command argument string: WRITE,READ,EXECUTE. When you want breaks on monitor references for all three break conditions, you can include one of the following command argument strings in the command line: either MUO,ALL or MUO,READ,WRITE,EXECUTE.

If you use SET BREAK with no conditions, the previously specified conditions are used. 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 Ored 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 you want to remove a break condition, the condition to be removed can be specified in the following command:

```
SET BREAK NO condition
```

If all existing break conditions are to be removed, the following command line can be issued:

```
SET BREAK NONE
```

This command will remove all existent break conditions, but will not remove a previously specified address. An example of 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
```

## SET BREAK Command (Cont.)

When a break occurs, one of the messages:

```
%ADDRESS BREAK AT USER PC xxxxxx
```

```
%ADDRESS BREAK AT EXEC xxxxxx UUU AT USER xxxxxx
```

will be printed, and your terminal will be left at monitor level. The second message is produced when MUUU was included in the SET BREAK command line. If you type:

```
.CONTINUE
```

the program will continue execution at the instruction that caused the break.

### Characteristics

The SET BREAK command:

Leaves your terminal at monitor level.

Can be used on KI10 and KL10 processors only.

Does not destroy your job's core image.

### Examples

1. Break if the contents of 1000 are read or written by the program.

```
.SET BREAK AT 1000 ON READ,WRITE (RET)
```

.

2. Break if the instruction is fetched from the default location, in this case 1000.

```
.SET BREAK EXECUTE (RET)
```

.

3. Remove existing break conditions.

```
.SET BREAK NONE (RET)
```

.

4. Break at any EXECUTE or WRITE monitor references.

```
.SET BREAK 1000 MUUU,EXECUTE,WRITE (RET)
```

.

## SET CDR Command

### Function

The SET CDR command sets the file name of the next file to be read by the card reader. This command is usually unnecessary, but allows you to change the order of jobs being spooled. The batch controller uses this command to read card decks.

### Format

SET CDR file-name

Where: **file-name** must be one to three characters; the extension is assumed to be .CDR.

### Characteristics

The SET CDR command:

Requires LOGIN.

Leaves your terminal at monitor level.

Destroys your core image.

### Example

Create a file for output.

```
.TYPE FOO.CDR (RET)
```

```
THIS IS THE OUTPUT
```

It may be necessary to spool the card reader.

```
.SET SPOOL CDR (RET)
```

The file FOO will be the next to be read by the card reader.

```
.SET CDR FOO (RET)
```

Type the output from the card reader on your terminal.

```
.TYPE CDR: (RET)
```

```
THIS IS THE OUTPUT
```

.

## SET CPU Command

### Function

The SET CPU command allows privileged users to change the processors where jobs can run. It is used in a multiprocessing system to specify whether the programs that 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:

- another SET CPU command with a different specification is given,
- a KJOB command is issued,
- a privileged 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 monitor call remains in effect until a RESET or EXIT monitor call or another SETUOO with a different specification is executed. When an EXIT or RESET monitor call is executed, the job reverts to the specification given in the last SET CPU command. When you log 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 you want to change the CPU specification.

### Formats

SET CPU CPxn

Adds the specified CPU to your job's CPU specification.

SET CPU NO CPxn

Removes the specified CPU from your job's CPU specification.

SET CPU ALL

Adds all of the CPUs to your job's CPU specification.



## SYSTEM COMMANDS AND PROGRAMS

### SET CPU Command (Cont.)

SET CPU ONLY CPxn

Changes the CPU specification so that it includes only the specified CPU.

Where: x is:

U designating a logical name,

I designating a KI10 processor,

L designating a KL10 processor,

or S designating a KS10 processor.

n is a decimal number from 0 to the number of processors in the system.

#### Characteristics

The SET CPU command:

Leaves your terminal at monitor level.

Does not destroy your job's core image.

#### Restrictions

The privileges required for using this command are determined by bit 5 (JP.CCC) of the privilege word, .GTPRV.

#### Examples

1. Your job can run only on CPU1.

```
.SET CPU ONLY CPU1 (RET)
```

.

2. Your job can run on CPUA.

```
.SET CPU CPUA (RET)
```

.

## SET DEFAULT BUFFERS Command

### Function

The SET DEFAULT BUFFERS command changes the number of buffers in your memory area. The default number of buffers is 6. Use this command to increase or decrease the number of memory buffers that your program will use.

A larger number of buffers will usually allow your program to run faster. A smaller number of buffers will allow you to run larger programs.

### NOTE

If you increase the number of buffers, it may interfere with the operation of old overlay programs and other large programs.

### Format

```
SET DEFAULT BUFFERS n
```

Where: n is the number of disk buffers you want your program to use. The argument is a decimal number between 0 and 511. When n is 0, the number of buffers is set to the system default, which is 6.

### Characteristics

The SET DEFAULT BUFFERS command:

Requires that you be logged in.

Does not destroy your core image.

Leaves your terminal at monitor level.

### Example

Set the number of buffers to 4.

```
.SET DEFAULT BUFFERS 4 (RET)
```

**SET DEFAULT PROTECTION Command****Function**

The SET DEFAULT PROTECTION command sets a default protection code for files you create explicitly with TECO or implicitly with PIP. Any value you set with this command overrides the system's standard protection code. Your system administrator sets the system's standard protection code, which is usually <055> or <057>.

**Formats**

```
SET DEFAULT PROTECTION <nnn>
```

```
SET DEFAULT PROTECTION ON
```

```
SET DEFAULT PROTECTION OFF
```

Where: <nnn> is the value that you can specify as the protection code for any files you might create.

**ON** tells the system to use the protection code you specified previously, even though you may have since used the SET DEFAULT PROTECTION OFF command.

**OFF** tells the system to use the system's standard protection code when you create files, even if you have issued a SET DEFAULT PROTECTION <nnn> command.

**Associated Commands**

**PRESERVE** - This command can be used to assign a protection code of <157> or <155> to any or all of existing files.

**PROTECT** - This command can be used to assign any protection code to any or all of existing files.

**Restriction**

SET DEFAULT PROTECTION affects only files created by TECO and by using the COPY command.

**Characteristics**

The SET DEFAULT PROTECTION command:

Leaves your terminal at monitor level.

**Example**

Create a file with TECO.

```
.MAKE PROTEC.TST RET
*EX ESC ESC
```

Use DIRECTORY to see the protection of the file.

```
.DIR PROTEC.TST RET
```

```
PROTEC TST 0 <055> 13-SEP-80 DSKC: [27,5434]
```

## SET DEFAULT PROTECTION Command (Cont.)

Change the default protection code.

```
.SET DEFAULT PROTECT <175> (RET)
```

Create another file with TECO.

```
.MAKE PROTEC.TS1 (RET)
*EX (ESC) (ESC)
```

DIRECTORY shows both files.

```
.DIR PROTEC (RET)

PROTEC TST 0 <055> 13-SEP-80 DSKC: [27,5434]
PROTEC TS1 0 <175> 13-SEP-80
```

Note the old protection and the new protection.

Turn the specified protection off.

```
.SET DEFAULT PROTECT OFF (RET)
```

Create a third file.

```
.MAKE PROTEC.TS2 (RET)
*EX (ESC) (ESC)
```

A DIRECTORY shows the 3 files.

```
.DIR PROTEC (RET)

PROTEC TST 0 <055> 13-SEP-80 DSKC: [27,5434]
PROTEC TS1 0 <175> 13-SEP-80
PROTEC TS2 0 <055> 13-SEP-80
```

Note the protection codes.

Turn the protection code on.

```
.SET DEFAULT PROTECT ON (RET)
```

Create a fourth file.

```
.MAKE PROTECT.TS3 (RET)
*EX (ESC) (ESC)
```

DIRECTORY shows the 4 files.

```
.DIR PROTEC (RET)

PROTEC TST <055> 13-SEP-80 DSKC: [27,5434]
PROTEC TS1 <175> 13-SEP-80
PROTEC TS2 <055> 13-SEP-80
PROTEC TS3 <175> 13-SEP-80
```

Note the protection codes.

## SET DENSITY Command

### Function

The SET DENSITY command sets a default density (bits/inch) for the specified magnetic tape. You must have the device assigned to your job to use SET DENSITY.

### Format

```
SET DENSITY dev:density
```

Where: **dev:** is the physical or logical name of the magnetic tape device for which the density is to be set. Refer to Section 1.8.1 for a description of device names.

**density** is one of the following:

```
200 bits/inch (8.1 rows/mm)
556 bits/inch (22.5 rows/mm)
800 bits/inch (32.2 rows/mm)
1600 bits/inch (65.3 rows/mm)
6250 bits/inch (255.5 rows/mm)
```

Both arguments are required.

### Characteristics

The SET DENSITY command:

Leaves your terminal at monitor level.

### Associated Messages

If the density argument is not compatible with the list above, an error message is printed:

```
?ILLEGAL DENSITY FOR DRIVE
```

### Example

Set the density of magnetic-tape unit 5 to 556.

```
.SET DENSITY MTA5:556 (RET)
```

## SET DSKFUL Command

### Function

The SET DSKFUL command controls the treatment of your job when it is attempting output and there is either not enough space available on the file structure being referenced or your quota for that structure is exceeded.

### Formats

#### SET DSKFUL ERROR

Stops output and an error condition is passed to the program. Most programs respond to the error condition by issuing an error message and returning your job to monitor level without any opportunity for you to continue.

#### SET DSKFUL PAUSE

Stops output and suspends execution of the program. An error message is printed on your terminal and control of the job is returned to the monitor. Generally, you should use the SEND command at this point to request assistance from the operator.

Execution of the program can be resumed with the CONTINUE command as long as you do not issue a command that 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 you specify otherwise as a switch to LOGIN.

### Characteristics

The SET DSKFUL command:

Leaves your terminal at monitor level.

### Example

.SET DSKFUL PAUSE

## SET DSKPRI Command

### Function

The SET DSKPRI command allows privileged users to set the priority for their 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 one 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. monitor call. (See the TOPS-10 Monitor Calls Manual.) The priority specified in the SET DSKPRI command remains in effect until one of the following occurs:

- Another SET DSKPRI command is given with a different priority.
- A KJOB command is issued.
- Your program overrides the SET DSKPRI command by issuing a DISK. monitor call with a different priority.

### Format

```
SET DSKPRI n
```

Where: n is a decimal number from -3 to +3 indicating the priority to be associated with your job's disk operations. When n is 0, the priority is the normal timesharing priority.

### Characteristics

The SET DSKPRI command:

Leaves your terminal at monitor level.

### 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. You are always allowed a 0 priority.

### Example

Set your priority to 2.

```
.SET DSKPRI 2 RET
```

.

**SET FLOATING POINT Command****Function**

The SET FLOATING POINT SIMULATION command takes the place of the following unsupported microcode:

Code	Description
UFA	Unnormalized Floating Add
DFN	Double Floating Negate
FADL	Floating Add Long
FSBL	Floating Subtract Long
FMPL	Floating Multiply Long
FDVL	Floating Divide Long

You should rewrite any programs you have that use this code.

If floating point simulation has been set, your attempt to use one of these instructions will be trapped by the monitor module KASER, which will process the instruction.

**Formats**

```
SET FLOATING POINT SIMULATION
SET FLOATING
SET F
```

These formats of the command enable floating point simulation.

```
SET FLOATING POINT NO SIMULATION
SET FLOATING NO
SET F P N
```

These formats disable the floating point simulation.

**Characteristics**

The SET FLOATING POINT command:

Returns your terminal at monitor level.

Does not destroy your core image.

**Example**

Floating point is enabled.

```
.SET F (RET)
```

Floating point is disabled.

```
.SET F P N (RET)
```



## SET HOST Command

### Function

The SET HOST command connects you to another host computer on the same network. If you are logged in, the system detaches your terminal from the node that received the command, and connects your terminal to the node you specified.

The node you specify must be a network node where you can run LOGIN or an equivalent program. The SET HOST command cannot be used to connect to a node that does not have a command interpreter. Use the NETWORK command with the /MCR switch to determine the nodes in the network that have command interpreters. MCR is a name that is used by the NETWORK program to signify a command interpreter.

Using the SET HOST command, you can remain at one terminal and switch the terminal connection from one system to a different system.

After issuing the SET HOST command, you can use the LOGIN or ATTACH command to gain access to the new host system.

Once a terminal has been connected to a particular system, it remains connected to that system until another SET HOST command is issued from that terminal. For example: you log in to a system, issue a SET HOST command to KS4101 (another host system), do your work, and then log off the system. Then, when another user uses your terminal, he too will be running on the KS4101 host system until he issues another SET HOST command or until the system is reloaded.

### Format

SET HOST node-id

Where: **node-id** is an identifier of the node to which you want to connect. It can be a node-name or a node-number.

### Characteristics

The SET HOST command:

Leaves your terminal at monitor level.

Does not require LOGIN.

Does not destroy your core image.

## SET HOST Command (Cont.)

### Associated Messages

If the specified node does not have a command interpreter, your job remains at your current node and the system prints the following message:

```
?NODE DOES NOT SUPPORT REMOTE TERMINALS
```

If the node you specified is not a recognizable node in the network, your job remains at your current node and the system prints the following message. This message would occur if the specified node was not on line, the link between the systems is down, or you misspelled the node name.

```
?UNDEFINED NETWORK NODE
```

If you omit the node-id argument, the system prints the following error message:

```
?NOT ENOUGH ARGUMENTS
```

### Examples

1. You transfer to node number 1.

```
.SET HOST 1 (RET)
```

2. You attempt to connect to node BJB22. The node-id is not that of a valid network node. The system returns control to monitor level of the node from which you issued the command and the system prints an error message:

```
.SET HOST BJB22 (RET)  
?UNDEFINED NETWORK NODE
```

3. You attempt to connect to a node that cannot interpret commands because it does not have a command decoder. The command causes an error message to be printed, and returns control to monitor level on the system where you issued the command.

```
.SET HOST DN78S (RET)  
?NODE DOES NOT SUPPORT A COMMAND DECODER
```

## SET HPQ Command

### Function

The SET HPQ command allows privileged users to place their jobs in a high-priority run queue. When your job is in a high-priority run queue, you can obtain a faster response than in the normal timesharing queues. The job remains in the specified high-priority queue until one of the following occurs:

- You issue another SET HPQ command specifying a different high-priority queue.
- You log out.
- Your program overrides the SET HPQ command by issuing an HPQ monitor call with a different value.

If an HPQ monitor call overrides the command, the level specified in the monitor call remains in effect until a RESET or EXIT monitor call or another HPQ monitor call with a different value is executed. When an EXIT or RESET monitor call is executed, the job is returned to the high-priority queue specified in the last SET HPQ command.

### Format

SET HPQ n

Where: n is a decimal number from 0 to 15 indicating the high-priority queue to be entered. When n is 0, the queue is the normal timesharing queue. Queue numbers from 1 to 15 are high-priority queues. The number of high-priority queues is an installation parameter and can be less than 15.

### Associated Messages

If you are not privileged, the system responds with an error message:

?NO PRIVS TO SET PRIORITY THAT HIGH

### Characteristics

The SET HPQ command:

Does not destroy your core image.

Leaves your terminal at monitor level.

### 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 high-priority queue you can attain.

**SET HPQ Command (Cont.)**

**Example**

You try to set your run queue to 4.

```
.SET HPQ 4 (RET)
```

```
?NO PRIVS TO SET PRIORITY THAT HIGH
```

Your job is placed in high-priority queue number 2.

```
.SET HPQ 2 (RET)
```

## SET PHYSICAL Command

### Function

The SET PHYSICAL command sets the maximum current physical page limit (CPPL) that your job can use, if the word LIMIT is included in the command line. (CPPL is described in the TOPS-10 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 when a program will go virtual.

### Format

```
SET PHYSICAL LIMIT nP
                GUIDELINE nK
```

Where:      LIMIT or GUIDELINE can be used in the command line.  
            The default is GUIDELINE.

K can be specified within the range 1 to 256K;   P can 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.

### NOTE

The monitor regards a limit as a set maximum number. When the monitor reaches the specified limit, it begins virtual handling. However, the GUIDELINE argument allows the monitor to use discretion in determining the page size.

### Characteristics

The SET PHYSICAL command:

Requires LOGIN.

Leaves your terminal at monitor level.

Does not destroy your core image.

SYSTEM COMMANDS AND PROGRAMS

SET PHYSICAL Command (Cont.)

Examples

1. Check your core memory assignment:

```
.CORE(RET)
VIRT. MEM. ASSIGNED 2P (CURRENT LIMIT: 512P MAX LIMIT: 512P)
PHYS. MEM. ASSIGNED 2P (GUIDELINE: 512P MAX LIMIT: 510P)
SWAP SPACE LEFT: 3669P
```

Change the physical guideline to 100 pages:

```
.SET PHYSICAL 100P(RET)
```

Show core assignment:

```
.CORE(RET)
VIRT. MEM. ASSIGNED 2P (CURRENT LIMIT: 512P MAX LIMIT: 512P)
PHYS. MEM. ASSIGNED 2P (GUIDELINE: 100P MAX LIMIT: 510P)
SWAP SPACE LEFT: 3671P
```

2. Show core memory assignment:

```
.CORE(RET)
VIRT. MEM. ASSIGNED 2P (CURRENT LIMIT: 512P MAX LIMIT: 512P)
PHYS. MEM. ASSIGNED 2P (GUIDELINE: 512P MAX LIMIT: 510P)
SWAP SPACE LEFT: 3484P
```

Set your physical limit to 100 pages:

```
.SET PHYSICAL LIMIT 100P(RET)
```

Show your memory assignment:

```
.CORE(RET)
VIRT. MEM. ASSIGNED 2P (CURRENT LIMIT: 512P MAX LIMIT: 512P)
PHYS. MEM. ASSIGNED 2P (CURRENT LIMIT: 100P MAX LIMIT: 510P)
SWAP SPACE LEFT: 3484P
```

## SET RETRY Command

### Function

The SET RETRY command governs the function of the magnetic tape drive controller DX10 and DX20. The controller performs soft error recovery from the magnetic tape without logging the soft errors in the ERROR.SYS file.

The error reporting function is set by SET RETRY MTxnnn:OFF. In this case, the monitor reports soft errors, logging them in the ERROR.SYS file.

### Formats

SET RETRY MTxnnn: ON

This format of the SET RETRY command reports soft errors. Errors are logged in the ERROR.SYS file.

SET RETRY MTxnnn: OFF

This format of the command restores the default condition.

Where: **MTxnnn:** is a physical or logical device name. This argument must be included in the command string.

### Characteristics

The SET RETRY command:

Leaves your terminal at monitor level.

#### NOTE

Magnetic tape errors are either hard or soft. Soft errors can be recovered by the controller and are due to inconsistencies in the hardware.

Hard errors are inconsistencies from which the controller cannot recover. Hard errors are always recorded in the ERROR.SYS file, terminating magnetic tape drive operation.

Pre-7.00 versions of TOPS-10 did not record soft errors. As a result, the controller could not record the number of soft errors encountered. The message after an UNLOAD command always had a 0 in the soft error count.

The SET RETRY command allows you to determine the number of soft errors that occurred in a magnetic tape read, since the errors are recorded in the ERROR.SYS file.

Version 7.00 of TOPS-10 enables the controller to count soft errors and record them to be read after an UNLOAD command.

**SET RETRY Command (Cont.)**

**Example**

Set the retry function:

.SET RETRY MTA2:

.



## SET SPOOL Command

### Function

The SET SPOOL command adds devices to or deletes devices from the current list of devices being spooled for your 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.

### Formats

SET SPOOL dev1:,dev2:,...,devn:

Adds the specified devices to your job's spool list.

SET SPOOL ALL

Places all spooling devices into the spool list.

SET SPOOL NONE

Clears the entire spool list.

SET SPOOL NO dev1,dev2,...,devn

Removes the specified devices from your job's spool list.

Where: **dev1:,dev2:,...,devn:** are the physical device names of one or more devices to be added to or deleted from the current spool list. These names can be taken from the following list: CDP:, CDR:, LPT:, PLT:, PTP:. Refer to Section 1.8.1 for device name formats.

### Characteristics

The SET SPOOL command:

Leaves your terminal at monitor level.

### Restrictions

To remove devices from your list of spooled devices, your job must have one of the following:

- The privilege bit set in .GTPRV.
- Bit 28 (200 octal) set in the STATUS word by the operator SET SCHED command.
- Logged in under [1,2].

## SYSTEM COMMANDS AND PROGRAMS

### SET SPOOL Command (Cont.)

#### Examples

1. Add the card punch to your list of spooled devices.

```
.SET SPOOL CDP: 
```

.

2. Delete the line printer from your list of spooled devices.

```
.SET SPOOL NO LPT: 
```

.

3. Clear your list of spooled devices.

```
.SET SPOOL NONE 
```

.

## SET TERMINAL or TERMINAL Command

### Function

The SET TERMINAL command (or TERMINAL command) is equivalent to the SET TTY command. Refer to the SET TTY command for a description of its function.

**SET TIME Command****Function**

The SET TIME command sets a CPU time limit for your job. When your job reaches the time limit, the job stops, and a message is printed on your terminal. You can continue your job by typing CONTINUE, but no time limit is in effect unless it is reset.

**Format**

```
SET TIME n
```

Where: n is the number of seconds of CPU time that the job is limited to. An argument of 0 turns the time limit off.

**Characteristics**

The SET TIME command:

Leaves your terminal at monitor level.

**Restrictions**

The SET TIME command is illegal in a batch job. A batch job has its time limit set by the /TIME switch in the QUEUE or SUBMIT command string or on the \$JOB card.

**Example**

Create a program with an infinite loop.

```
.MAKE LOOP.FOR (RET)
*I10  CONTINUE (RET)
      GO TO 10 (RET)
      END (RET)
(ESC) (ESC)
*EX (ESC) (ESC)
```

Compile and load the program.

```
.LOAD LOOP (RET)
FORTRAN: LOOP
MAIN.
LINK:   LOADING
EXIT
```

Set the time limit to 5 seconds.

```
.SET TIME 5 (RET)
```

Clear the incremental run time, so that the SET TIME command can be checked.

```
.TIME (RET)
11.08
11.28
KILO-CORE-SEC=95
```

SYSTEM COMMANDS AND PROGRAMS

**SET TIME Command (Cont.)**

Start the program.

```
.START (RET)  
?TIME LIMIT EXCEEDED (RET)
```

As expected, the time limit was exceeded.

```
.TIME (RET)  
5.00  
16.08  
KILO-CORE-SEC=134  
  
.
```

## SET TTY or TTY Command

### Function

The SET TTY command (or TTY command) declares properties of the terminal on which the command is typed. The word TERMINAL can be used instead of the word TTY in the SET TTY command.

### Defaults

The system has a terminal default for any terminal that has not been declared to be of a particular type. This default is generic TTY. Before you specify the characteristics of your terminal, the TTY defaults are in effect. The exceptions to this are terminal input and output speeds.

To discover the characteristics that the system has assigned to your terminal, use the INITIA TTY command. This is illustrated in the Example.

If you use the SET TTY TYPE command, the system assumes defaults according to the terminal type you declared. These defaults, and the defaults for TTY, are listed in Table 2-2.

SYSTEM COMMANDS AND PROGRAMS

SET TTY or TTY Command (Cont.)

Table 2-2  
Terminal Hardware Characteristics

T Y P E	L O W E R C A S E	T A B S	F O R M	W I D T H	D I S P L A Y	F I L	P A G E	A L T M O D E
DAS21	Y	Y	Y	80	Y	0	24	N
LA120	N	Y	Y	132	N	0	N/A	N
LA30	N	N	N	72	N	1	N/A	N
LA34	Y	Y	N	132	N	0	N/A	N
LA36	Y	N	N	132	N	0	N/A	N
LA38	Y	Y	N	132	N	0	N/A	N
TTY	N	N	N	72	N	0	N/A	N
TTY33	N	N	N	72	N	1	N/A	Y
TTY35	N	Y	Y	72	N	1	N/A	Y
VT05	N	Y	N	72	Y	2	20	N
VT06	N	N	N	72	Y	1	25	N
VT100	Y	Y	N	80	Y	0	24	N
VT50	Y	Y	N	80	Y	0	12	N
VT52	Y	Y	N	80	Y	0	24	N
VT61	Y	Y	N	80	Y	0	24	N
2741	Y	N	N	128	N	3	N/A	N

NOTE

The page length is set by default for video terminals only. Therefore, the notation N/A (not applicable) appears in the table under PAGE for hard-copy terminals.

## SYSTEM COMMANDS AND PROGRAMS

### SET TTY or TTY Command (Cont.)

#### Formats

SET TTY arg

Sets a characteristic for your terminal.

SET TTY NO arg

Deletes a characteristic for your terminal.

Where: **arg** is the characteristic that can be set by this command. The arguments to SET TTY are listed below.

**NO** turns off the bits set by the argument it accompanies. For example, SET TTY LC sets lowercase capability. SET TTY NO LC turns off lowercase ability.

The word SET is an optional portion of the command line.

Argument	Function
ALTMODE	Converts the ALTMODE codes of 175 and 176 to the ASCII standard ESCape character 033. NO ALTMODE restores the individual identities of the codes 175 and 176.
BLANKS	Controls the output of blank lines. NO BLANKS suppresses the output of consecutive RETURNS. This is useful for display terminals, because NO BLANKS will conserve space on the screen, allowing more output to be displayed at one time.
COPY	Controls the output of the ESCape character to a VT5x terminal. Refer to the <u>VT52 Owner's Manual</u> .
CRLF	Controls the automatic RETURN at the end of the line. Used with the WIDTH argument, CRLF gives an automatic RETURN at the edge of the terminal screen. This feature does not work at monitor level. NO CRLF suppresses the automatic RETURN at the edge of the screen. The default setting is CRLF.
DEBREAK	Notifies the system that the terminal has a feature that allows the computer to lock the keyboard. (Model 2741 terminals.)
DEFER	Suppresses echoing to a video terminal until output to the terminal is finished. This is used when you type characters before the output is finished. DEFER holds the characters until the output is done; NO DEFER allows them to be echoed on your terminal before the output is done.



## SYSTEM COMMANDS AND PROGRAMS

### SET TTY or TTY Command (Cont.)

Argument	Function
DISPLAY	Notifies the system that the terminal you are using is a display terminal. Programs use this information to control output to the terminal.
ECHO	Controls echoing to the terminal. Local-copy terminals (terminals that automatically print each character as you type it) do not require that the system echo characters to the terminal. Use NO ECHO for a local-copy terminal. ECHO is the default setting.
ELEMENT nnn	Changes the typing element number for Model 2741 terminals. The element numbers available are:

Element	Meaning
987	APL correspondence
029	Standard correspondence
097	Call 360 BASIC
963	Extended binary
938	BCD
988	APL (EBCD)

SET TTY ELEMENT also sets NO LC.

FILL n  
Assigns filler class n to the terminal. Some terminals require one or more filler characters to be sent following certain control characters such as line feed (LF) and horizontal tab (HT).

Table 2-3 illustrates the number of fillers sent for each character and filler class. The filler characters are CR (carriage-return), and DEL (DELETE or RUBOUT) for all other characters.

No fillers are supplied for image mode output. The default is determined by the declared terminal type.

SYSTEM COMMANDS AND PROGRAMS

SET TTY or TTY Command (Cont.)

Table 2-3  
Fill Characters

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 <sup>1</sup>
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 <sup>2</sup>
automatic CR <sup>3</sup>		0	2	4	4
CRLF on output	015-012	0	3	9	9 <sup>4</sup>
XON	021	1	1	1	1
TAPE	022	1	1	1	1
XOFF	023	1	1	1	1
NTAP	024	1	1	1	1

<sup>1</sup> 1 if 0-3 spaces to tab stop; 2 if 4-7 spaces to tab stop.  
<sup>2</sup> Output only; no fillers on input.  
<sup>3</sup> Refer to the SET TTY CRLF command.  
<sup>4</sup> Sum of the fillers output for a CR and LF.

**FORM** Controls the output of line-feeds for the formfeed and vertical tab characters. NO FORM instructs the system to output the line-feeds. FORM is a declaration that the terminal has the capability of generating the line-feeds. The default is determined by the declared terminal type.

**GAG** Controls the reception of messages that were sent with the SEND command. GAG instructs the system to prevent messages from reaching the terminal when the job is at user level. NO GAG allows messages to reach the terminal at any time. The default setting is GAG.

**HOLD** Controls the use of the SCROLL key on VT5x terminals. Refer to the VT52 Owner's Manual for instructions on the use of this key.

## SYSTEM COMMANDS AND PROGRAMS

### SET TTY or TTY Command (Cont.)

Argument	Function
LC	<p>Controls the translation of lowercase letters to uppercase letters. NO LC allows all characters to be translated to uppercase by the monitor.</p> <p>Frequently, it is convenient to have a terminal with both uppercase and lowercase simulate the behavior of one with uppercase only. TTY NO LC 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 output, you can determine whether translation was performed by the monitor. The default condition is determined by the declared terminal type.</p>
PAGE n	<p>Holds output to your terminal after n lines. You can also use CTRL/S and CTRL/Q after using this command. Refer to Section 1.5. CTRL/S stops output, CTRL/Q allows output to continue. If you use SET TTY PAGE with no argument or with the argument of 0, the system does not hold output, but you can use CTRL/S and CTRL/Q to control output. The default argument is determined by the declared terminal type.</p> <p>This argument is useful for display terminals because it prevents output from scrolling off the screen. When output is stopped by the system after n lines, the terminal bell rings. Use CTRL/Q to continue output to your terminal.</p> <p>NO PAGE turns off the ability to stop output using CTRL/S. After you use this command, output to your terminal is not stopped by the system (as with PAGE n) and cannot be stopped by CTRL/S.</p>
RTCOMPATIBILITY	<p>Disables the CTRL/R and CTRL/T features. (Refer to Section 1.5 for a description of CTRL/R and CTRL/T.) NO RTCOMPATIBILITY turns the CTRL/R and CTRL/T features on.</p>
SPEED m n	<p>Sets the terminal receiving speed to m and the transmitting speed to n. To set the same transmitting and receiving speed, use the m argument alone. You must adjust the speed of your terminal to match m and n using switches on your terminal.</p> <p>(This feature is not available on KS systems.) NO SPEED is an invalid argument.</p>

**SET TTY or TTY Command (Cont.)**

Argument	Function																								
TAB	Specifies that this terminal has hardware TAB stops every eight columns. The default is set according to the declared terminal type. When you use NO TAB, the monitor simulates tabs by sending the necessary number of SPACE characters.																								
TAPE	Controls the function of CTRL/S and CTRL/Q. TAPE causes CTRL/S to function like XOFF and CTRL/Q to function like XON when reading paper tape. Refer the TOPS-10 Monitor Calls Manual. NO TAPE causes CTRL/S and CTRL/Q to take their normal function. Refer to Sections 1.5.7 and 1.5.8.																								
TIDY	Specifies (for Model 2741 terminals) that every character occupy one print space. The terminal normally prints characters so that they appear the same as you type them in. For example, [ prints as ^<. In TIDY mode, [ prints as <.																								
TYPE nnnn	<p>Sets terminal characteristics defined by the declared terminal type (nnnn).</p> <p>The following terminal types are recognized by the system:</p> <table border="0" style="margin-left: 40px;"> <tr> <td>DAS21</td> <td>LA120</td> <td>LA34</td> <td>LA36</td> </tr> <tr> <td>LA38</td> <td>VT100</td> <td>VT61</td> <td>VT05</td> </tr> <tr> <td>VT06</td> <td>VT50</td> <td>VT52</td> <td>TTY33</td> </tr> <tr> <td>TTY35</td> <td>LA30</td> <td>2741</td> <td>TTY</td> </tr> </table> <p>TTY is a generic terminal and is the default for terminals whose types have not been declared. The characteristics set for these types are described in Table 2-2.</p> <p>The characteristics that are set by declaring the terminal type are:</p> <table border="0" style="margin-left: 40px;"> <tr> <td>LOWERCASE</td> </tr> <tr> <td>TABS</td> </tr> <tr> <td>FORM</td> </tr> <tr> <td>WIDTH</td> </tr> <tr> <td>DISPLAY</td> </tr> <tr> <td>FILL</td> </tr> <tr> <td>PAGE</td> </tr> <tr> <td>ALTMODE</td> </tr> </table> <p>TYPE is not a necessary portion of the command string. For example, you can set your terminal type to that of an LA120 by typing SET TTY LA120.</p>	DAS21	LA120	LA34	LA36	LA38	VT100	VT61	VT05	VT06	VT50	VT52	TTY33	TTY35	LA30	2741	TTY	LOWERCASE	TABS	FORM	WIDTH	DISPLAY	FILL	PAGE	ALTMODE
DAS21	LA120	LA34	LA36																						
LA38	VT100	VT61	VT05																						
VT06	VT50	VT52	TTY33																						
TTY35	LA30	2741	TTY																						
LOWERCASE																									
TABS																									
FORM																									
WIDTH																									
DISPLAY																									
FILL																									
PAGE																									
ALTMODE																									

## SYSTEM COMMANDS AND PROGRAMS

### SET TTY or TTY Command (Cont.)

Argument	Function
UC	Equivalent to SET TTY NO LC. Translates all lowercase letters to uppercase. NO UC is equivalent to SET TTY LC.
WIDTH n	Sets the terminal width (the length of the line) to n characters. This command combined with the SET TTY CRLF command enables you to specify the widest possible margin for your terminal. The default setting is determined by the declared terminal type.

#### Characteristics

The SET TTY command:

Leaves your terminal at monitor level.

Does not require LOGIN.

#### Restrictions

The SET TTY (or TTY) command is not available to batch jobs.

#### Example

Tell the system that your terminal is a VT05.

```
.SET TTY VT05 RET
```

Issue the INITIA TTY command. The system responds with the terminal characteristics of a VT05 terminal.

```
.I TTY RET
```

```
RJ052A KL #1026/1042 08:24:08 TTY52 SYSTEM 1026/1042
CONNECTED TO NODE KL1026(26) LINE #52
JOB 33 USER MAROTTA,M [27,5434]
DSKC: KL1026 SYSTEM DISK DSKC:
DSKB: KL1026 SYSTEM DISK DSKB:
TYPE:VT05  NOLC      TABS          NOFORM
ECHO:DEFER CRLF     WIDTH:72    NOGAG
DISPLA     FILL:2   NOTAPE     PAGE:20
NOBLANKS   NOALMOD   RCVSPEED:150 XMTSPEED:2400
NODEBREA   NOTIDY   NORTCOMP   NOSLAVE
NOHDX      REMOTE
```

## SET VIRTUAL LIMIT Command

## Function

The SET VIRTUAL LIMIT command sets the current virtual page limit (CVPL). CVPL is described in the TOPS-10 Monitor Calls Manual.

## Format

```
SET VIRTUAL LIMIT nK
                    nP
```

Where: K can be specified within the range 1 to 256K; P can be specified within the range 1 to 512p.

1K equals 1024 words and 1P equals a page of 512 words. If K and P are omitted, K is assumed.

If you use 0 for the argument, the value of CVPL is assumed. CVPL is set by the system administrator and is the default virtual page limit.

LIMIT is an optional portion of the command line.

## Characteristics

The SET VIRTUAL LIMIT command:

Leaves your terminal at monitor level.

## Example

Set your page limit to 32 pages.

```
.SET VIRTUAL LIMIT 32P (RET)
```

Run the SORT program:

```
.R SORT (RET)
```

```
*TEAM.RNO=TEAM.RNO/RECORD:80/KEY:1:80
/COLLATE:FILE:COLLATE.RNO (RET)
```

System message says that you set your page limit too low.

```
?SRTNEC NOT ENOUGH CORE FOR SORT/MERGE
```

```
*^C
```

Reset page limit.

```
.SET VIRTUAL LIMIT 132P (RET)
```

```
.R SORT (RET)
```

```
*TEAM.RNO=TEAM.RNO/RECORD:80/KEY:1:80
/COLLATE:FILE:COLLATE.RNO (RET)
```

**SET VIRTUAL LIMIT Command (Cont.)**

Program ran successfully in the 132 pages that you specified.

[SRTXPN EXPANDING TO 130P]  
SORTED 14 RECORDS  
34 KEY COMPARISONS,  
2.43 PER RECORD  
16 RECORD LEAVES IN MEMORY  
0 RUNS  
0:00:00 CPU TIME, 16.71 MS PER RECORD  
0:00:00 ELAPSED TIME  
\*^C

## SET WATCH Command

### Function

The SET WATCH command tells the system to print incremental job statistics automatically. This command provides you with a tool for measuring the performance of your programs.

### Formats

SET WATCH arg,arg,...

Prints the specified WATCH statistics. You specify the statistics using the arguments.

SET WATCH ALL

Prints all the WATCH statistics.

SET WATCH NONE

Stops printing all WATCH statistics.

SET WATCH NO arg,arg,...

Stops printing the specified WATCH statistics.

Where: arg can be one of the following arguments.

The following arguments enable printing whenever a monitor command passes the job from monitor level to user level.

DAY Prints the time of day, as [hh:mm:ss]

VERSION Prints the version of the program in standard format (refer to the VERSION command). The version is printed when your job issues a GET or RUN command, a SETNAM monitor call, or if a DAEMON wait occurs.

The following arguments enable printing whenever the terminal is returned to monitor level through the CTRL/C, EXIT, HALT, ERROR IN JOBn, or DEVICE xxx OPR zz ACTION REQUESTED messages.

READ Prints the incremental number of disk blocks read modulo 4096.

RUN Prints the incremental runtime.

WAIT Prints the wait time. (That is, the time elapsed since you started or continued your program.)

WRITE Prints the incremental number of disk blocks written modulo 4096.



## SYSTEM COMMANDS AND PROGRAMS

### SET WATCH Command (Cont.)

When an UNLOAD command or UUO is executed,

MTA Prints magnetic tape statistics in the form:

```
[MTxnnu:reelid READ (C/H/S) = a/b/c WRITE (C/H/S) = d/e/f]
```

Where: x represents the tape controller.

nn is the node number.

u represents the drive number.

reelid is the reel identification.

a is the number of characters read.

b is the number of hard-read errors.

c is the number of soft-read errors.

d is the number of characters written.

e is the number of hard-write errors.

f is the number of soft-write errors.

When a, b, and c are 0, the information pertaining to READ will not be printed.

When d, e, and f are 0, the information pertaining to WRITE will not be printed.

When any files are accessed,

FILES Prints a message for each file accessed. The message takes the form:

```
[Fxn: dev:file.ext[ppn,sfd,...]errorz]
```

Where: F (where it appears) signifies a FILOP.

x = L for Lookup

E for Enter

R for Rename

n is the channel number

z is the error if the LOOKUP, ENTER, or RENAME failed.

Error codes can be found in the TOPS-10 Monitor Calls Manual.

Any combination of the arguments can be specified in any order. Statistics are not printed for commands that do not run programs, such as ASSIGN or PJOB. When you log in, your job is set to WATCH the statistics for which you notified the system manager. The information on what statistics to WATCH is kept in ACCT.SYS. The default is MTA.

**SET WATCH Command (Cont.)****Associated Messages**

If you type SET WATCH with no arguments, or with an invalid argument, the following error message occurs:

```
?ARGS ARE: DAY,RUN,WAIT,READ,WRITE,VERSION,MTA,FILES,ALL,NONE
```

**NOTE**

Enabling WATCH output interacts with the incremental data printed by the TIME and DSK commands.

**Characteristics**

The SET WATCH command:

Leaves your terminal at monitor level.

**Example**

List the arguments:

```
.SET WATCH 
?ARGS ARE: DAY,RUN,WAIT,READ,WRITE,VERSION,MTA,FILES,ALL,NONE
```

Set the statistics for time, run time, wait time, version number:

```
.SET WATCH DAY 
.SET WATCH RUN 
.SET WATCH WAIT 
.SET WATCH VERSION 
```

Run a program:

```
.RUN LOOP 
[14:26:40]
?LOOP.SAV NOT FOUND
[0.06 0.91 ]

.R PIP 
[14:26:50]

[S:PIP 33B(260) + ]
*C
[0.08 6.08 ]

.R TECO 
[14:27:03]

[S:TECO 230(162) + ]
*C
[2.05 6.10 ]

.
```

## SETSRC Program

### Function

You can use the SETSRC program to change your job search list or the system search list. A search list is the order of the file structures that are to be searched whenever you implicitly or explicitly specify the generic device DSK:. This search list is originally defined by the system manager to include the file structures that you can access. With the SETSRC program, you can alter the search list by adding or deleting file structures.

The search list is of the form:

```
fsl:, fs2:, fs3:, FENCE, fs4:
```

Where: fs is the name of the file structure.

The file structures on the left of the FENCE form the active search list and represent the generic device DSK: for your job. The active search list is a list of the file structures that the monitor will search and the order in which the file structures are to be searched. The search goes from left to right.

The files to the right of the FENCE comprise the passive search list and represent the file structures that were once in the active search list. File structures are kept in the passive search list so that quotas can be checked on a DISMOUNT or KJOB command; these structures are not searched. The FENCE is the boundary between the active and passive search lists.

You can change the job's search list by adding or deleting file structures, using the MOUNT and DISMOUNT commands. Because the SETSRC program does not create a UFD if one does not exist, the MOUNT command should be used to create a UFD. (See the SETSRC specification in the TOPS-10 Software Notebooks for a complete description of the SETSRC program.)

### Format

R SETSRC

When the program prompt (\*) appears, you can respond with any of the following commands:

Command	Function
A	<p>Adds one or more file structures to the existing active 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:</p> <ol style="list-style-type: none"> <li>1. If no asterisk appears in the specifications (for example, fsl, fs2) or if an asterisk appears before the file structure names (for example, *,fsl, fs2), the file structures are added to the end of the search list.</li> </ol>

## SYSTEM COMMANDS AND PROGRAMS

### SETSRC Program (Cont.)

Command	Function
A (Cont.)	
	2. If the asterisk follows the file structure names (for example, fs1, fs2, *), the file structures are added to the beginning of the search list.
	3. If the asterisk appears in the middle of the file structures (for example, 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.
C	Creates a new search list for this job. Any file structures in the current search list that are not in the new list are moved to the passive search list.
CP	Creates a new default directory path.
CS	Creates a new system search list (that is, the file structure search list for device SYS:). You must be logged in under [1,2] to use this command.
H	Prints information (HELP) about the available commands.
M	Modifies 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	Removes 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. Ersatz devices are not affected by this command.
T	Prints the job search list.
TP	Prints the default directory path.
TS	Prints 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 name that they modify. Other switches can appear anywhere in the command string. The switches can be abbreviated as long as the abbreviation is unique.

## SETSRC Program (Cont.)

### Switches That Modify File Structures

/CREATE	Allows new files to be created on the file structure.
/NOCREATE	Does not allow new files to be created on the file structure when DSK: is specified, but allows files to be superseded. Files can be created on the file structure if you specify the file structure name explicitly.
/NOWRITE	Does not allow writing on the file structure for this job (that is, the file structure is read-only).
/WRITE	Allows 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 is equivalent to /NOWRITE.

### Switches That Modify The Directory Path

(use only with the CP command)

These switches can be typed in directly as commands by omitting the CP command and the slash (for example, SCAN is equivalent to CP/SCAN).

/NOSCAN	Cancels the scan switch for the directory path.
/SCAN	Sets the scan switch for the directory path.

### Switches That Modify the DSK: or SYS: Specification

(use 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 (for example, NOSYS is equivalent to M/NOSYS).

/LIB:[proj,prog]	Sets the job's library directory to the UFD [proj,prog] and adds it to your DSK: specification. This means that if a file is not found in your directories in your search list, the library directory will then be searched for the file.
/NOLIB	Removes the library directory from your DSK: specification.
/NOSYS	Removes the SYS: specification from your DSK: specification.
/NONEW	Removes the [1,5] directory from your SYS: specification.

**SETSRC Program (Cont.)**

/SYS Adds the SYS: specification to your DSK: specification. If a file cannot be found in your directories in your search list or in your library directory (if /LIB:[proj,prog] has been specified), the system directory [1,4] will then be searched for the file.

/NEW Adds the directory [1,5] to your 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 SETSRC program:

Places your terminal at user level.

Destroys your core image.

**Restrictions**

You must be logged in under [1,2] to create a new system search list.

**Example**

Your search list is defined as DSKB:.

```
.R SETSRC (RET)
*T (RET)
*DSKB:, FENCE
```

Add DSKA: to the end of the search list.

```
*A DSKA: (RET)
```

Your search list is defined as DSKB:,DSKA:.

```
*T (RET)
DSKB:,DSKA:,FENCE
```

Add DSKC: to the beginning of the search list.

```
*A DSKC:,* (RET)
```

```
*T (RET)
```

```
DSKC: ,DSKB: ,DSKA: ,FENCE
```

Remove DSKA: from the search list.

```
*R DSKA: (RET)
```

```
*T (RET)
```

```
DSKC: ,DSKB: ,FENCE,DSKA:
```

## SETSRC Program (Cont.)

Disable writing on DSKB:.

```
*M DSKB:/NOWRITE 
```

Set your library directory to [27,500] and add it to your DSK: specification.

```
*M/LIB:[27,500] 
```

Add SYS: to your search list.

```
*SYS 
```

```
*T 
```

```
/LIB:[27,500]/SYS DSKC:,DSKB:/NOWRITE,FENCE,DSKA:
```

Your DSK: and SYS: specifications are listed first, followed by your search list.

```
*TS 
```

The system search list is defined as DSKA:,DSKB:,DSKC:.

```
*DSKA:,DSKB:,DSKC:
```

.

## SKIP Command

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

### Formats

SKIP MTxnnu: x FILES

Advances forward x files.

SKIP MTxnnu: x RECORDS

Advances forward x records.

SKIP MTxnnu: EOT

Advances forward to the logical end of tape.

Where: **MTxnnu:** is the physical device name of the magnetic tape unit. Device names are discussed in Section 1.8.1.

**x** is a decimal number representing the number of records or files to skip over.

### Characteristics

The SKIP command:

Leaves your terminal at monitor level.

Destroys your core image.

### Example

Space the tape MTA2: forward 2 files:

.SKIP MTA2: 2 FILES **RET**



**START Command****Function**

The START command begins execution of a program that you loaded previously with either the LOAD or GET command, or that you interrupted while running (for example, with a CTRL/C).

You can optionally specify an explicit start address. If omitted, the address supplied in the file (.JBSA) is used. If you specify an address argument and the job was executing a monitor call when interrupted (that is, it was at monitor level but not in TTY input wait or SLEEP mode), the monitor continues the job at the location at which it was interrupted and eventually traps to the specified START address.

If you try to START an execute-only program at a specified address, you receive an error message. If the high segment of this program is sharable, execution of the START command turns on the user-mode write protect bit.

**Format**

START addr

Where: **addr** is the octal address where execution starts if other than the location specified within the file (.JBSA). This argument is optional. If you do not specify this argument, the address comes from .JBSA. You can specify a starting address of 0.

**Characteristics**

The START command:

Places your terminal at user level.

Does not function during device I/O.

Requires memory.

**Example**

Type a program.

```
.TYPE PROG.FOR (RET)
TYPE 69
69 FORMAT (' TESTING EXECUTION')
END
```

Load the program.

```
.LOAD PROG (RET)
FORTRAN:PROG
MAIN.
LINK:LOADING
EXIT
```

**START Command (Cont.)**

SAVE the program:

.SAVE PROG   
PROG SAVED

START the program.

.START

TESTING EXECUTION

END OF EXECUTION

CPU TIME: 0.01 ELAPSED TIME: 0.03

EXIT

## SUBMIT Command

### Function

The SUBMIT command places 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-spec,log-file-spec/switches
```

### Format

```
SUBMIT jobname=control-file-spec,log-file-spec/switches
```

Where: **jobname** is the name of the job being entered into the queue. If you omit the job name, the default is the name of the log file.

**control-file-spec** is the name of the input file. This file contains all monitor-level and user-level commands to be processed by the batch controller. This argument is required. If you omit the file extension of the control file, the system looks for a file with the extension .CTL.

**log-file-spec** is the name of the log file. The batch controller uses this file to record its processing of the job. If you omit the log file-spec, the default will be the name of the control file with the extension .LOG.

Only these two files can be specified in a request to the batch input queue.

To print a list of input queue entries on your terminal, type the SUBMIT command with no arguments. To print a list of entries made by a specific PPN, type the SUBMIT command followed by the PPN in brackets. Refer to examples.

**/switches** specify options to the command. The switches are listed in the description of the QUEUE command.

### Characteristics

The SUBMIT command:

Leaves your terminal at monitor level.

Does not require LOGIN when only a list of the queue entries is desired.

SYSTEM COMMANDS AND PROGRAMS

**SUBMIT Command (Cont.)**

**Example**

Submit the file PAROL.CTL to be processed with time limit of one hour.

```
.SUBMIT PAROL/TIME RET  
[INP02:PAROL=/SEQ:44/TIME:01:00:00/CORE:50P
```

Show the request in the batch queue:

```
.SUBMIT RET
```

```
INPUT QUEUE:  
STS      JOB      SEQ      PRIO      TIME      CORE      USER  
---      ---      ---      ---      ---      ---      ---  
RUN      PARIL    44      10      01:00:00  50P      MAROTTA
```

## SYSTAT Command

### Function

The SYSTAT command prints status information about the system.

To write the information on the disk as a file with the name SYSTAT.TXT, assign device DSK: with logical name SYSTAT.

SYSTAT prints the status of the system: system name, time of day, date, uptime, percent null time (idle plus lost time), number of jobs in use.

SYSTAT prints the status of each job logged-in: job-number; project-programmer number ([OPR] is the operator's job, [SELF] is your job), terminal line number (CTY is console terminal, DET is detached, Pn is PTY number); program name being run; program size; job and swapped state (refer to the TOPS-10 Monitor Calls Manual), and runtime since the job logged in.

SYSTAT prints the status of high segments being used: name (PRIV is nonsharable, OBS is superseded); device or file structure name from which the segment came; the directory name; the size (SW is swapped out, SWF is swapped out and fragmented, F is in core and fragmented on disk, SPY signifies using the SPY monitor call); the number of users in core or on the disk.

SYSTAT prints the amount of swapping space used, the virtual memory used, swapping ratio, active swapping ratio, virtual memory saved by sharing, and average job size.

SYSTAT prints status of busy devices: device name, job-number, how device is assigned (AS is ASSIGN command, INIT is INIT or OPEN monitor call, AS+INIT is both ways).

SYSTAT prints system file structures: free blocks, mount count, single-access structures, and private structures.

SYSTAT prints non-network dataset control: number of the terminal, status of the terminal.

### Format

SYSTAT arg

Where: **arg** is one or more single letters (in any order) used to specify any subset of the SYSTAT output. The argument is optional. The following is a list of the arguments:

Argument	Function
B	Prints busy device status.
C	Prints continuous SYSTAT.
D	Prints dormant segment status.
E	Prints non-disk error report.
F	Prints file structure status.
G	Prints other system status.
H	Prints help text listing the arguments.
J	Prints job status.
L	Lists the SYSTAT output on LPT.

## SYSTAT Command (Cont.)

## Argument Function

N Prints non-job status (that is, all information except J).  
 P Prints disk performance.  
 S Prints short job status.  
 T Prints dataset status.  
 U Includes user names in output.  
 V Prints paged output for display terminals.  
 X Reads the file DSK:CRASH.EXE if found, otherwise reads specified crashed monitor written in .EXE format.

## Meanings of job state codes:

AU Disk alter UFD wait.  
 ^C Job stopped.  
 CA Core allocation wait (to be locked).  
 CB Disk core block scan wait.  
 CW Command wait.  
 ^D DAEMON wait.  
 D1 DECTape control wait.  
 D2 2nd DECTape control wait.  
 DA Disk storage allocation (SAT block) wait.  
 DC Data control wait.  
 DI Disk I/O wait.  
 DS Disk I/O wait satisfied.  
 EQ ENQ/DEQ resource wait  
 EV Exec virtual memory wait.  
 EW Event wait.  
 HB Hibernate state.  
 IO I/O wait.  
 JD DAEMON wait.  
 MM Memory management resource wait.  
 NA Nap (short sleep).  
 NU Null state.  
 OW Operator wait.  
 PI Paging I/O wait.  
 PS Paging I/O wait satisfied.  
 RN In a run queue.  
 SL Sleep wait.  
 ST Stop (^C) state.  
 SW Swapped out.  
 SWF Swapped out and fragmented on disk.  
 TI TTY I/O wait (input).  
 TO TTY output.  
 TS TTY I/O wait satisfied.  
 ^W Command wait.  
 WS I/O wait satisfied.

You can obtain output for individual jobs by specifying one of the following after the command:

A number n that causes information to be listed only for the specified job (that is, job n). A period causes information for your job to be printed.

A project-programmer number specified in square brackets causes information to be printed only for jobs with the specified project-programmer number. The project and/or programmer number can be specified with an asterisk.

SYSTEM COMMANDS AND PROGRAMS

**SYSTAT Command (Cont.)**

A number preceded by a number sign (#n) causes information to be printed only for jobs from the indicated terminal (that is, 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 your terminal at monitor level.

Destroys your core image.

Does not require LOGIN.

**Example**

.SYSTAT<sup>(#1)</sup>

STATUS OF R2366A KL #1026/1042 AT 10:03:10 ON 02-AUG-79

UPTIME 1:18:07+1:18:07, 45+33%NULL TIME = 44+32%IDLE + 1+1%LOST, 11+17%OVHD  
60 JOBS IN USE OUT OF 200. 59 LOGGED IN, 13 DETACHED OUT OF 200.  
(LOGMAX)

JOB	WHO	LINE#	WHAT	SIZE (P)	STATE	RUN TIME
1	[OPR]	P1J8	DTELDLDR	11+1	SL	1
2	[OPR]	CTY	OPSER	2+6	HB	1
3	[OPR]	DET	DAEMON	26+SPY	SL	7 01
4	10,2100	DET	PPN	1	^C	7
5	[OPR]	1	SYSTAT	15+SPY	^C	38
6	30,5347	56	DIRECT	6+22	DI	10
7	10,3076	200	INITIA	6+7	^C	1:01
8	[OPR]	4	OPSER	2+6	HB	5
9	2,5	DET	NETLDR	16+9	EW	0 \$
10	[OPR]	P2J2	ORION	36+16	SL	6
11	[OPR]	P3J8	OPROMO	23+14	SL	2
12	10,5324	154	SNOBOL	4+21	^C SW	4
13	[OPR]	P0J2	QUASAR	52+16	SL	6
14	[OPR]	P4J8	OPR	35+16	HB	2
15	30,5221	121	DDT	28	TI	4
16	[OPR]	DET	PULSAR	30+16	HB	29
17	[OPR]	13	DEL	2+15	TI	24 #
18	30,5226	54	SYSTAT	23+SPY	^C	3 \$
19	[OPR]	P21J8		27+16	EW	0
20	[OPR]	DET	FILDAE	28	HB	3
21	[OPR]	DET	MIC	2+13	SL	7
22	10,64	45	FAST	29+13	^C	15
23	10,10	224	SYSTAT	23+SPY	^C SW	30 \$
24	[OPR]	P6J2	LPTSPL	31+16	HB	7
25	77,4655	30	SOS	11+16	TI	8
26	[OPR]	P10J2	BATCON	34+16	SL	8
27	[OPR]	P11J2	CDRIVE	11+16	SL	0
28	30,4571	124	QUERY	59+46	^C SW	4
29	[OPR]	P13J2	SPROUT	12+16	^C SW	0
30	[OPR]	P14J2	SPRINT	17+16	HB	0

SYSTEM COMMANDS AND PROGRAMS

SYSTAT Command (Cont.)

JOB	WHO	LINE#	WHAT	SIZE (P)	STATE	RUN TIME
31	[OPR]	DET	SYSINF	10	HB	0
32	[OPR]	P16J2	JUSTIF	42+28	HB	0
33	[OPR]	DET	PERF	15+SPY	SL	18
34	[OPR]	DET	NIK	5+15	EW	8
35	[OPR]	P15J2	ACTDAE	23+16	HB	1
36	30,5250	215	NETLDR	15+8	TI	2 #
37	30,5232	213	DTECO	9+14	TI	30
38	50,75	113	DT	9+14	TI	33 #
39	10,5057	P7J44	ENQDEQ	16+1	SL	11 #
40	10,5442	11	DIRECT	4+22	^C SW	8
41	[SELF]	64	QUEUE	14+19	^C	2
42	30,5454	57	S	9+12	^C SW	27
43	30,4515	55	SIRUS	29+79	TI	9 #
44	10,5057	212	OPSER	2+6	HB	16
45	[OPR]	6	BACKUP	13+15	TI	2:09
46	30,5431	214	BACKUP	36	TI	10
47	30,5171	127	FILCOM	2+4	TI	8
48	[OPR]	DET	NIK	5+15	TO	14
49	27,5107	110	QUEUE	13+19	^C	31
50	10,5057	P12J44	ENQDEQ	16+1	SL	11 #
51	30,5051	122	SIRUS	29+79	TO	22 #
52	[OPR]	0	OPR	37+16	HB	2
53	10,5057	P20J44	ENQDEQ	16+1	RN	11 #
54	10,2162	211	DIRECT	4+22	^C	1
55	10,5324	DET	BLKXFR	4	TI	1 &
56	10,5057	DET	SYSTAT	23+SPY	^C SW	2 \$
57	10,176	216	SPRINT	29	TI	12
58	[OPR]	DET		1	^C SW	0
59	50,3615	2	QSRCRS	68+16	TI	6
60	[SELF]	P5J26	SYSTAT	18+SPY	RN	1 \$

# MEANS NON-SYSTEM HI-SEG

& MEANS LOCKED IN CORE

\$ MEANS EXECUTE ONLY N MEANS JOB RUNS IN HPQ N

JNN IS THE CONTROLLING JOB, PNN CORRESPONDS TO TTY437+NN



SYSTEM COMMANDS AND PROGRAMS

SYSTAT Command (Cont.)

HIGH SEGMENTS:

PROGRAM	DEVICE	OWNER	HIGH(P)	USERS
INITIA	DSKC	SYS	7	1
OPSER	DSKC	SYS	6	3
GLXLIB	DSKC	SYS	16	10
DTELDR	DSKC	SYS	1	1
(PRIV)		JOB 11	14	1
MIC	DSKC	SYS	13	1
BACKUP	DSKC	SYS	15	1
JUSTIF	DSKC	SYS	28	1
FAST	DSKC	SYS	13	1
NIK	DSKC	SYS	15	2
(PRIV)		JOB 35	16	1
ENQDEQ	DSKB	10,5057	1	3
DIRECT	DSKC	SYS	22	3
SOS	DSKC	SYS	16	1
QUEUE	DSKC	SYS	19	2
NETLDR	DSKC	SYS	9	1
SIRUS	DSKC	1,5	79	2
FILCOM	DSKC	SYS	4	1
LIBO11	DSKC	SYS	46 SW	1
(PRIV)		JOB 19	16	1
NETLDR	DSKC	1,3	8	1
SNOBOL	DSKC	SYS	21 SW	1
(PRIV)		JOB 42	12	1
(PRIV)		JOB 59	16	1
DTECO	DSKC	SYS	14	1
DEL	DSKC	10,5277	15	1
DT	GAL0	50,10	14	1

SWAPPING SPACE USED = 515/8200 = 6%

VIRT. CORE USED = 1574/8200 = 19%

117P CORE LEFT

ACTIVE SWAPPING RATIO = 151/1691 = 0.1

VIRT. CORE SAVED BY SHARING = 315/(315+1574) = 17%

AVERAGE JOB SIZE = 1118/60 = 18.6P+771/60 = 12.9P TOTAL=1889/60 = 31.5P

SYSTEM COMMANDS AND PROGRAMS

**SYSTAT Command (Cont.)**

BUSY DEVICES:

DEVICE	JOB	WHY	LOGICAL
TTY144	1	INIT	
TTY146	1	INIT	
TTY147	1	INIT	
MPX1	1	INIT	
CDR260	27	INIT	
LPT260	24	AS+INIT	
LPT261	24	AS+INIT	
LPT263	24	INIT	
DTA266	0	AS	
DTA267	0	AS	
'LA260	16	AS	
'LA261	16	AS	
MTB260	45	AS	
'LB263	16	AS	
MTB264	19	AS+INIT	
'LB264	16	AS	
'LC260	16	AS	
'LC261	16	AS	
MTD261	18	AS	
'LD261	16	AS	
TSK26	34	INIT	

SYSTEM FILE STRUCTURES:

NAME	FREE	MOUNT	
GALO	27872	2	
DSKH	113270	1	
TAPE	15782	0	PRIVATE STRUCTURE
GMU0	85992	1	
BLKA	83880	1	
DSKA	2381	25	
BUNK	149687	2	
BLKX	38802	1	
BLKK	106400	2	
DSKC	27740	60	
DSKN	60710	5	
DSKB	74150	60	
SIRS	32738	2	PRIVATE STRUCTURE
BLKY	103792	0	
TOTAL FREE	923196		

## TECO Command

### Function

The TECO command opens a file for editing using TECO. See the TECO Manual in the TOPS-10 Software Notebooks.

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

### Format

TECO dev:file.ext[directory] core

Where: **dev:** is the device or file structure name containing the existing file. If omitted, DSK: is assumed.

**file.ext** is the file name and the file name extension of the existing file. If omitted, the arguments of the last MAKE or TECO command are used.

**[directory]** is the directory name where the file is stored. If omitted, your default directory is assumed.

**core** is the amount of core you want to be assigned to the job.

You can pass switches to TECO by preceding the switch with a slash in the TECO command string. When COMPIL interprets the command string, it passes the switches to TECO.

### Characteristics

The TECO command:

Places your terminal at user level.

Destroys your core image.

### Associated Commands

MAKE - Creates a file with TECO.

### Example

Edit a file named WONDER.ALG.

```
.TECO WONDER.ALG (RET)
[5K CORE]
*EX (ESC) (ESC)
```

## SYSTEM COMMANDS AND PROGRAMS

### TIME Command

#### Function

The TIME command prints the total run time since the last time you issued the TIME command, followed by the total run time used by the job since it was initialized (logged-in), followed by the integrated product of running time and memory size (KILO-CORE-SEC=). Time is printed in the following format:

hh:mm:ss.hh

Where:    hh is hours  
          mm is minutes  
          ss.hh is seconds to nearest hundredth.

Interrupt level and job scheduling times are charged to whichever user was running the system job when the interrupt or rescheduling occurred.

#### NOTE

If automatic run time is enabled with the SET WATCH RUN command, the incremental run time is usually 0.

#### Format

TIME job

Where:    job is the job-number of the job whose time is desired. If job is omitted, the job to which the terminal is attached is assumed. In this case, the monitor prints the incremental run time (run time since the last TIME command) as well as the total run time since the job was initialized.

#### Characteristics

The TIME command:

Leaves your terminal at monitor level.

Does not require LOGIN when requesting time for another user's job.

Does not destroy core image.

## SYSTEM COMMANDS AND PROGRAMS

### TIME Command (Cont.)

#### Example

The command is given for the first time after LOGIN. Therefore, the incremental time equals the total time since LOGIN.

```
.TIME (RET)
21.36
21.36
KILO-CORE-SEC=175
```

Show that the time is reset:

```
.TI (RET)
0.00
21.36
KILO-CORE-SEC=175
```

Run the DIRECT program:

```
.DIR/F (RET)

WONDER.REL      DSKC: [27,4072]
WONDER.QOR
PROG2.QLG
PROG2.REL
NEW.
ADD1.BAS
OLD.BAS
FACT.BAS
NAME.BAS
WONDER.ALG
WONDER.FOR
TEST.ALG
PROGL.FOR
TESTER.
016DAE.TMP
002DAE.TMP
LOOP.F4
LOOP.REL
FILEC.DAE      DSKB:
```

The DIRECT command took .70 seconds of run time and 183 kilo-core-seconds.

```
.TI (RET)
0.70
22.06
KILO-CORE-SEC=183
```

## TPUNCH Command

### Function

The TPUNCH command is used to place entries into the paper tape punch output queue. This command is equal to the following form of the QUEUE command:

```
QUEUE PTP:jobname=file-spec/switches,file-spec/switches,...
```

### Format

```
TPUNCH jobname=file-spec/switches,file-spec/switches,...
```

Where: **jobname** is the name of the job being entered into the queue. The default is the name of the first file in the request.

**file-spec** is the file specification of the files to be punched. You can use the wildcard construction in the file-spec. File specification is described in Section 1.8.

**/switches** are options that further describe the request. The switches are listed in the description of the QUEUE command.

You can have the entries in the paper tape punch queue listed on your terminal by typing TPUNCH with no arguments. You can see the entries made by a specific project-programmer number by typing TPUNCH followed by the project-programmer number in brackets.

### Characteristics

The TPUNCH command:

Leaves your terminal at monitor level.

Destroys your core image.

Does not require LOGIN when only a list of queue entries is desired.

## TPUNCH Command (Cont.)

## Example

Punch the file SNDMP.REL on paper tape. Provide three copies of the output and punch it in binary mode.

```
.TPUNCH SNDMP.REL/TAPE:BINARy/COPIES:3 RET
[PTP:SNDMP=/SEQ:77/LIMIT:500/TAPE:BINARy/COPIES:3]
```

Show all the paper tape requests in the queue:

```
.TPUNCH RET

OUTPUT QUEUES:
DEV      JOB      SEQ      PRIO      LIMIT      USER
----      ---      ---      ----      -
PTP      SNDMP  77       10        500        MAROTTA
/TAPE:BINARy/COPIES:3
```

Kill the request:

```
.TPUNCH SNDMP=/KILL RET
[ONE JOB KILLED]
```

**TYPE Command****Function**

The TYPE command prints the contents of the specified file(s) on your terminal. If more than one file is requested in the command string, the files are printed one after another without an indication of the beginning and ending of a file.

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

**Format**

TYPE file-spec

Where: **file-spec** is one or more file specifications separated by commas. The file name (including any extension) is required. File specification is discussed in Section 1.8.

If you omit the device name, the default is DSK:. If you omit the directory, the default is the directory you are currently logged in to.

The full wildcard construction can be used for the file name and 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 your terminal at monitor level.

Destroys your core image.

**Example**

```
.TYPE WONDER.ALG RET
BEGIN  INTEGER S, C, B, A;
!THIS PROGRAM WILL ADD THREE NUMBERS AND;
!PRINT THE RESULT;
      WRITE (" [2 C] WHAT ARE THE NUMBERS TO ADD [1 C] ");
      READ (S);
      READ (C);
      READ (B);
      A:=C+S+B;
      WRITE (" [2 C] THE SUM OF ");
      PRINT (S,3,3);
      WRITE (" AND ");
      PRINT (C,3,3);
      WRITE ("AND ");
      PRINT (B,3,3);
      WRITE (" IS: ");
      PRINT (A,3,3);
      WRITE (" [2 C] ");
END
```



## UNLOAD Command

### Function

The UNLOAD command rewinds and unloads a tape, either magnetic tape or DECTape. When magnetic tape is unloaded, WATCH statistics are printed on the operator's terminal. These statistics also print on your terminal by default. To turn off the printing, use the SET WATCH NO MTA command.

### Format

UNLOAD dev:

Where: **dev:** is a magnetic tape (MTxnn) or a DECTape (DTxnn).  
Device names are discussed in Section 1.8.1.

### Characteristics

The UNLOAD command:

Leaves your terminal at monitor level.

Runs the PIP program.

Destroys your core image.

### Examples

1. Unload DECTape unit 7.

```
.UNLOAD DTA7: RET
```

.

2. Unload magnetic tape unit 3.

```
.UNLOAD MTA3: RET
```

.

**USESTAT Command****Function**

The USESTAT command (CTRL/T) returns status information about your job. This status information is printed on three lines in eight columns. The information printed includes:

- incremental daytime in seconds
- incremental run time in seconds
- incremental disk reads
- incremental disk writes
- program name
- core size
- job state
- program counter (PC)

Refer to Section 1.5.9 for a description of the status information returned as a result of the USESTAT command (or CTRL/T).

The results obtained from the USESTAT command can also be obtained by typing CTRL/T. CTRL/T can be typed at user level as well as at monitor level. You must have your terminal set to RTCOMP. Refer to the SET TTY RTCOMP command.

**Formats**

USESTAT

(CTRL/T)

**Characteristics**

CTRL/T:

Does not interrupt program execution.

Does not echo on your terminal.

USESTAT:

Leaves your terminal at monitor level.

**Example**

```
.USESTAT RET
DAY: 16.56 RUN: 0.00 RD:0 WR:0 PIP 2+5K ^C PC:400173
```

**VERSION Command****Function**

The VERSION command prints the version number of the program in your core area (that is, the last program you ran implicitly or explicitly). One use of this command is to determine the program that printed an error message on your terminal. If your terminal is still at user level (that is, a character other than a period was printed) after the message, you can type the following:

```
^C          (two CTRL/Cs if the program is not waiting for
            input)
.VERSION
```

The monitor returns with the name of the program in core (that is, the one presumed to have printed the message) and the version number of that program. After receiving the information, you can type CONTINUE to return your terminal to user mode. If the message was a fatal message (that is, a period was printed after the message), you do not have to type CTRL/C because the terminal is already at monitor level. In most cases, you cannot type CONTINUE 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 forms:

[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.
[+]	A GETSEG monitor call has been performed to a high segment that matches the low segment.
[+ high]	A GETSEG monitor call has been done to a high segment that does not match the low segment.
[blank]	The high segment has been released.
[+ FROM file-spec]	The file has been read into memory.
[+ high NOT SHARABLE FROM file-spec]	The file read into memory is not sharable.

With the VERSION command, the low and high segments are represented in the format:

```
name version
```

## SYSTEM COMMANDS AND PROGRAMS

### VERSION Command (Cont.)

With the SET WATCH VERSION command, the low and high segments are represented in one of three formats:

name version	The program is not from SYS:.
:name version	The output is the result of a SETNAM monitor call (for example, 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 the name of the program and the version is in standard format. When the version number is output, the standard format is:

major-version minor-version(edit)-group who modified program

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 is DEC development, 1 is all other DEC personnel, and 2-7 is customer). 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 last modified program 1.
7(5)	major version 7, minor version 0, edit number 5, group that last modified program 0.
54A	major version 54, minor version A, edit number 0, group that last modified program 0.

DAEMON is called on execution of the VERSION command if the first page of the high segment is paged out.

#### Format

VERSION

#### Characteristics

The VERSION command:

Leaves your terminal at monitor level.

Does not destroy your job's core image.

## VERSION Command (Cont.)

## Examples

1. Run the TECO program.

```
.R TECO 
```

Halt the program.

```
*^C
```

Find version number.

```
.VERSION   
TECO 23B(162) +
```

Continue the program and exit.

```
.CONTINUE   
*EX  
```

2. Type a file.

```
.TYPE WONDER.ALG   
BEGIN  INTEGER S, C, B, A;  
! THIS PROGRAM WILL ADD THREE NUMBERS AND;  
! PRINT THE RESULT; ^C
```

Find version number of PIP, which executes TYPE.

```
.VERSION   
PIP 33B(260) +
```

Run and halt the SYSTAT program.

```
.R SYSTAT 
```

```
STATUS OF      R5743A SYS #40/2 AT 14:45:44 ON 22-APR-75  
UPTIME 7:36:20+7:34:33, ^C  
^C
```

Find the version number of SYSTAT.

```
.VERSION   
SYSTAT 472(156)
```

**WHERE Command****Function**

The WHERE command allows you to determine the system to which a specified device is connected. The information given includes:

- node name
- node number
- software-identification
- creation-date of software

**Format**

WHERE devn:

Where: **devn:** is the physical device name. OPR: allows you to find the location of the operator's controlling terminal. The colon (:) in the device name is optional.

**Associated Messages**

Output is in the form:

```
NODE node-name system-identification system-creation-date
specified-device
```

If the system does not recognize the device you specified, it prints the message:

```
?NO SUCH DEVICE
```

**Characteristics**

The WHERE command:

Does not require LOGIN.

Leaves your terminal at monitor level.

Does not destroy your core image.

**Examples**

1. To find the location of a specific device LPT262:, type the command:

```
.WHERE LPT262: RET
NODE KL1026(26) R2364A KL # 1026 07-18-80 LPT262
```

This example shows that the node name is 1026. System identification is RZ364A KL # 1026/1042. Creation date of software is 07-18-80. The specified device is LPT262.

SYSTEM COMMANDS AND PROGRAMS

WHERE Command (Cont.)

2. To find the physical node of your job's terminal, type the command:

```
.WHERE TTY:(RET)
NODE NOVA(31) DN87S V21(117) 04-04-80 TTY70 LINE #70
```

This example shows that the terminal is connected to node NOVA(31). The version number is V21(117). The creation date was 04-04-80. Your terminal number and line number are 70.

3. To find the physical node and terminal line of the operator's terminal, type:

```
.WHERE OPR:(RET)
NODE KL1026(26) RZ345A KL # 1026/1042 04-04-80 TTY4 LINE #4
```

4. To display the location of the system controlling terminal, type:

```
.WHERE CTY:(RET)
NODE KL1026(26) RZ345A KL # 1026/1042 04-04-80 CTY LINE #7
```

## ZERO Command

### Function

The ZERO command clears the directory in the specified file structure. ZERO deletes all the files in the specified directory. The files cannot be recovered.

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

### Format

ZERO dev:[directory]

Where: **dev:** is a DEctape or a disk. This argument is required.

**[directory]** is a directory you have access to, if you specify a disk structure. If you do not specify a directory, your directory is assumed.

### Characteristics

The ZERO command:

Leaves your terminal at monitor level.

Destroys your core image.

### Examples

1. Delete the files on DEctape 4.

```
.ZERO DTA4:␣
```

.

2. Delete all files in your search list.

```
.ZERO DSK:␣
```

.

3. Delete all files in your SFD named CBDDT.

```
.ZERO DSKB:[27,4072,CBDDT]␣
```

.



## CHAPTER 3

### SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

The system informs you of the state of your job through diagnostic messages, error codes, and other system messages.

#### 3.1 DIAGNOSTIC MESSAGES

Diagnostic messages are printed on your terminal by the command interpreter. They occur when you type something that the system does not recognize or cannot use. Diagnostic messages are preceded and followed by a question mark.

For example, if you type a group of characters that the system is unable to recognize, the command interpreter intercepts the characters and prints them on your terminal, preceded and followed by a question mark.

##### Example

```
.PRINTFILE   
?PRINTFILE?  
.
```

A diagnostic message may also occur when you type a response that the system cannot use. For example, if you type a group of alphabetical characters when the system needs a numeric argument, the whole command is printed on your terminal, preceded and followed by question marks.

##### Example

```
.CORE ABC   
?CORE ABC?  
.
```

When you receive a diagnostic message, retype the command correctly.

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### 3.2 ERROR MESSAGES

Error messages are in two forms:

? A question mark at the beginning of the message indicates a fatal error message.

% A percent sign at the beginning of the message indicates an advisory, or warning message.

Error messages are typed on your terminal by the particular program that detected the error. You can find the program that detected the error message by typing CTRL/T. CTRL/T duplicates the USESTAT command, typing the state of your job and the program that you are currently running.

### 3.3 OTHER SYSTEM MESSAGES

Some system messages are output for your information. These messages are enclosed in brackets ([ ]).

#### Example

```
.[structure IN:  iiii  OUT:  ooooo USED:  nnn]
```

In the above example, the message was produced after the system checked disk usage for the user's area. This message is output by the system for the user's information.

### 3.4 FORMAT OF ERROR MESSAGES

System error messages are often preceded by a cryptic abbreviation of the error message and the program that output the message.

#### Example

```
[LGTRDU RECOMPUTING DISK USAGE ON structure-name]
```

In the above example, the abbreviated message is LGTRDU. Since the format of the error messages differs from one program to another, the abbreviated messages have been deleted from the messages as they are listed in Section 3.4. The messages are arranged alphabetically, for ease of reference. Where the message begins with a variable (such as job number, device name, or file name), the message is alphabetized by the first word in the constant text of the message. The variable parts of the error messages are printed in lowercase letters; constant text is printed in uppercase.

Each error message is accompanied by an explanation of its meaning, and, in parentheses, the program or command that produces the message.

Some error messages are preceded by a number in parentheses.

#### Example

```
?(1) NO DIRECTORY FOR PROJECT-PROGRAMMER NUMBER
```

These messages are documented in the TOPS-10 Monitor Calls Manual. The number (1) is an error code, used for reference to this documentation.

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### 3.5 SYSTEM MESSAGES

?1+lnK 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). You should wait a few minutes, and then attempt to log in again. If this message still appears, it should be reported to the operator.

m+n/p CORE  
VIR.CORE LEFT = v

Where: m is number of blocks in low segment.

n is number of blocks in high segment.

p is 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 is number of K blocks unassigned in core and on the swapping device.

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.

?nK BLOCKS OF CORE NEEDED

Your current core allocation is less than the contents of .JBFF.

?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)

ACCOUNTING SYSTEM FAILURE...

A program could not append an entry to the accounting file. Notify the operator. (LOGIN, LOGOUT)

?ACCOUNTING SYSTEM FAILURE DAEMON UOO FAILED DAEMON NOT RUNNING  
?ACCOUNTING SYSTEM FAILURE DAEMON UOO FAILED error code no.

An error occurred in the system accounting files, and the DAEMON UOO failed. Call the operator if you receive such a message. If the operator receives the message, continue the system for system debugging.

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?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(.JBPFI)); in other words, the address lies outside the bounds of your 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 (ASSIGN).

%ALREADY AT NODE xxnnnn

You are already logically established at the node you specified (xxnnnn). (SET HOST)

unit-id ALREADY MOUNTED ON unit

You tried to mount a device that was already mounted, or you requested that a mounted file structure be mounted on a different unit. (MOUNT)

?AMBIGUOUS ABBREVIATION:nn

A command or switch (nn) has been abbreviated to the point that it is not unique. (COMPIL)

?dev: APPARENT UFD PROTECTION FAILURE

Your attempt to delete a file has failed because of a UFD protection. You must have access privileges to delete a file. (KJOB)

?ARGS ARE: DAY, RUN, WAIT, READ, WRITE, VERSION, MTA, FILES, ALL, NONE

You 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 your job. (ASSIGN)

?ASSIGNED TO JOB n(1), n(2), ...

You specified a device of which none are available. The jobs listed own the devices. Wait until a device is free, then try again. (ASSIGN)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?ASSIGNED TO YOUR JOB

The device you specified is already assigned to your job.  
(ASSIGN)

?ATTACH TO USER JOB FAILED

DAEMON could not attach to your 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 you try to write on a write-locked tape or use a DECTape that has never been written on. (MONITOR)

?BOOTSTRAP LOADER IS NOT IN COPY; TRY /L

An attempt was made to write the bootstrap loader onto a DECTape with the /T switch before the loader was loaded into a core buffer and preserved with the COPY core image. (COPY)

?BOOTSTRAP LOADER WILL NOT FIT IN 3 BLOCKS

Your bootstrap loader is too big to fit into blocks 0, 1, and 2 of the output DECTape. (COPY)

?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 (that is, 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. (MAKLIB)

%CANNOT FIND FILE, (0) DSK:file-spec

The specified file does not exist in your default directory. Specify the area the file is stored in. (CREF)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### %CANNOT OVERRIDE PPN IMPLIED BY DEVICE

You specified a project-programmer number in a file specification that is in conflict with the project-programmer number associated with the device. For example, the file specification, NEW:FILE.EXT[12,144], would generate this message since "NEW:" implies the project-programmer number [1,5]. The project-programmer number specified by you is ignored. (FILE)

### ?CANNOT PROCESS EXTERNAL SYMBOLS

External symbols were encountered while loading the bootstrap loader with the /switch. (DTCOPY program)

### ?CANNOT PROCESS HIGH SEG'S

While loading the bootstrap loader with the /switch, high segment code was encountered. (DTCOPY program)

### ?CANNOT SPOOL A PHYSICAL DEVICE

The device that you specified cannot be spooled. Assign the device to your job, with the ASSIGN or MOUNT command. (SET SPOOL)

### ?CAN'T ACCESS QUEUE - 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)

### ?CAN'T ACCESS QUEUE - file structure name MUST NOT BE LOCKED OR WRITE PROT.

The named structure is being used to queue requests to the operator and therefore may not be locked or write-protected. SETSRC may be used to change the protection. (MOUNT, DISMOUNT, FILE)

### ?CAN'T ACCESS SYSTEM FILES

ACCT.SYS could not be read. Only the operator may LOGIN until ACCT.SYS is ready. Consult the operator. (LOGIN)

### ?CAN'T ADD TO YOUR FILE STRUCTURE SEARCH LIST n

n is the error code from STRUOO when trying to add a file structure to search list. (LOGIN)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?file-structure name CAN'T ADD TO YOUR FILE STRUCTURE SEARCH LIST n

The error code (n) was returned from the STRUOO, which tried to add this structure to your search list. See the STRUOO UOO error codes in the TOPS-10 Monitor Calls Manual. This is a system error and therefore is not a direct fault of the user. It is printed for potential diagnostic use. Consult the system administrator. (MOUNT)

?CAN'T ATTACH 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 CAN'T 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)

%CAN'T CLEAR SOFTWARE WRITE-LOCK ON STRUCTURE structure

You had structure software write-locked, and the STRUOO to clear this condition failed. This message is usually found by a %LGTURF message. (LOGOUT)

?CAN'T CONTINUE

The job was terminated due to:

1. An ERROR IN JOB message (except for HALT).
2. The EXIT UOO.
3. The CLOSE command.
4. The REA command when the device was INITed, and you attempted to continue your program at the point at which I/O was terminated. The job cannot be continued. (CONTINUE, CCONTINUE)

?CAN'T DECIPHER THAT

There is a syntax error in the command string. (UMOUNT)

?CAN'T ENTER OUTPUT FILE n file-descriptor

The ENTER to write the output file failed; n is the disk error code. (DUMP)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?CAN'T EXPAND TABLE xxxx

The DUMP program ran out of core in attempting to expand the indicated table. (DUMP)

?CAN'T FIND FILE file.ext

The specified file could not be found. (SCAN)

?CAN'T FIND INPUT FILE n file-descriptor

DUMP cannot locate the file specified as the input file; n is the disk error code. (DUMP)

?CAN'T FIND STRLST.SYS

The system is unable to locate or access STRLST.SYS, the system file that contains necessary information about the structures known to the system. Consult the system administrator. (MOUNT)

?CAN'T GET CORE FOR CLOCK TABLE

Extra core is needed, but is not available. (DAEMON)

?CAN'T GET SWAPPING PARAMETERS

DAEMON tried to obtain the job's swapping parameters and failed. (DAEMON)

?CAN'T GET SWAPPING POINTER FOR JOB

DAEMON tried to obtain the pointer to your job on the swapping space and could not because the GETTAB UO failed. (DAEMON)

?CAN'T GET USERS PPN

DAEMON tried to obtain your project-programmer number and could not because a GETTAB UO failed. (DAEMON)

?CAN'T LOOKUP DIRECTORY

The structure you specified was not a system library device. Specify a valid library structure. (HELP)

?CAN'T OPEN file-structure-name

The file structure is mounted but cannot be opened. No UFD is created, though one may already exist. (LOGIN)

?file-structure-name CAN'T OPEN

The specified device cannot be OPENed; either because it does not exist or because the OPEN UO failed. (UMOUNT)



## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?CAN'T OPEN DEVICE dev

The specified device does not exist or it is assigned to another user. (DAEMON)

?CAN'T OPEN INPUT DEVICE

The device you specified was not a system library. Specify only valid library structures. (HELP)

?CAN'T OPEN SWAP UNIT abc

DAEMON attempted to use the indicated swapping unit and failed. (DAEMON)

?CAN'T READ STRLST.SYS

A read error occurred when the system tried to read STRLST.SYS, the system file that contains necessary information about the structures known to the system. Consult the system administrator. (MOUNT)

?file structure name CAN'T REDEFINE FILE STRUCTURE n

"n" is the error code returned from the STRUOO which is attempting to redefine the structure to include the user. See the STRUOO UOO error codes in the TOPS-10 Monitor Calls Manual. This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. Consult the system administrator. (MOUNT)

?file structure name CAN'T REMOVE FILE STRUCTURE FROM SEARCH LIST n

"n" is the error code returned from the STRUOO which is attempting to remove the structure from your search list. See STRUOO UOO error codes in the TOPS-10 Monitor Calls Manual. This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. Consult the system administrator. (DISMOUNT)

?CAN'T RENAME-FILE PRESERVED

An attempt was made with the /DISPOSE:RENAME switch to delete a preserved file (that is, a file whose owner's field is greater than 0). (QUEUE)

?CAN'T SET OUR SEARCH LIST

DAEMON tried to set its search list and failed in its attempt. (DAEMON)

?CAN'T SET SEARCH LIST = USER'S

DAEMON attempted to set its file structure search list to be the same as your search list. (DAEMON)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?COMMAND ERROR--TYPE /H FOR HELP

General error response for most commands. The syntax of the command is in error and the command cannot be deciphered.

?COMMAND SWITCH REQUIRED

The given command string requires a MAKLIB command code. (MAKLIB)

?COMMAND SYNTAX ERROR  
TYPE /H FOR HELP

An illegal command string was entered. (GLOB)

?COMMA REQUIRED IN DIRECTORY

A directory name has been specified without the separating comma. (DUMP, QUEUE)

n COMMANDS IN QUEUE

The request you typed 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)

?CONFLICTING SWITCHES IN KJOB COMMAND

Conflicting switch-types have been specified to the KJOB command. (LOGOUT)

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)

%CPUn OPR ACTION REQUESTED

The job's CPU specification includes a CPU, that 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)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?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)

### ?DATA ERROR ON DEVICE PTR

A read error has occurred on the paper-tape reader. (PIP)

### ?DESTINATION DEVICE ERROR

An I/O error occurred on the output device. (GLOB)

### ?DETACH UO FAILED

DAEMON could not detach itself from the TTY. DAEMON does not detach itself if it is loaded with the debugging device. (DAEMON)

### ?DEVICE CAN'T BE REASSIGNED

1. The job's controlling terminal cannot be reassigned.
2. The logical name would be duplicated.
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 nnn: DOES NOT EXIST

The specified device (nnn:) does not exist. Try another device, or specify a generic device name. (EOF)

### ?DEVICE ERROR ON OUTPUT DEVICE

A write error has occurred on the output file. (MAKLIB)

### ?DEVICE FIELD MUST BE SPECIFIED

The format of the DEALLOCATE command requires that you specify the device name. (DEALLOCATE)

### ?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 DTCOPY command string is the DECTape. (DTCOPY)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?DEVICE NOT ASSIGNABLE

A non-privileged user cannot assign the requested device because it belongs to the restricted pool of devices. You should try to assign the device with the MOUNT command. (ASSIGN)

### ?DEVICE nnn: NOT AVAILABLE

The specified device cannot be initialized because another user is using it or because it does not exist. (EOF)

### ?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). (MAKLIB)

### 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 non-error messages. (UMOUNT)

### DON'T 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 file name, or two colons appeared in a row. For example, LPT:PTP: or DSKA::FILEX. (DUMP, QUEUE)

### ?DOUBLE DIRECTORY ILLEGAL

Two directory names cannot appear without an intervening file name. (DUMP, QUEUE)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?DOUBLE EXTENSION ILLEGAL

Two extensions cannot appear without an intervening file name or comma. (DUMP, QUEUE)

### ?DOUBLE FILE NAME ILLEGAL

Two file names appeared in a row or two periods appeared in a row; for example, Q TEST1 TEST2 or TESTX.MAC. This message can also occur when the wildcard construction is used illegally. (DUMP, QUEUE)

### ?DSK CAN'T BE REASSIGNED

An attempt was made to reassign the prototype disk device data block (DDB).

### ?DSKCHR FAILURE n ON UNIT abc

The DSKCHR UUO 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 (that is, there is no room for the file to be written on the DEctape).

### ?ENTER FAILURE n

The output file name is null; n is the error code for an illegal file name (nondisk devices only). (MAKLIB)

### ?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, name

The entry block of the named program is too large for the MAKLIB entry table, which allows for 1000 entry names. MAKLIB can be reassembled with a larger table. (MAKLIB)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?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 your 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.

?ERROR PARSING MESSAGE

You must end the message with either ESCape or RETURN. (PLEASE)

?ERSATZ DEVICE device

An attempt was made to remove an ersatz device from the search list with the setsrc command R. (SETSRC)

?EXCEED LOG-OUT m QUOTA BY n BLOCKS

The total number of blocks for all the user's files exceeds the maximum permitted value (mmm) by the indicated amount (nnn). If you are the operator, the DISMOUNT proceeds. Otherwise, this message is followed by the message "?DISMOUNT INCOMPLETE". Then, you may use PIP or the DELETE command to remove files. But, until the number of blocks is under the limit, you cannot dismount a structure or log out. (UMOUNT)

?EXECUTION DELETED

A program cannot execute 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; for example, 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. Check all files in your area for errors. (LOGIN)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?FILE NAME ALREADY IN USE

The specified file already exists. (COMPIL)

?FILE NAME REQUIRED FOR INPUT QUEUE

A file cannot be entered into the Batch input queue without a file name. (QUEUE)

?file name FILE NOT FOUND

The specified file cannot be located.

?FILE n NOT IN SAV FORMAT

You indicated with 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)

?FILE REQUEST NOT QUEUED

Your file request was not queued. Other messages give the reasons. (FILE)

?FILE R ILLEGAL IF DSK: CONTAINS NO UFD

You tried to do a FILE R command with a search list that has no device that is eligible to receive the recalled files. You should have at least one structure in your active search list that contains both a UFD and the status of /CREATE and /WRITE. You should adjust your search list with the SETSRC command and retry the FILE command. (FILE)

%FILES ARE DIFFERENT

The two input files specified in the command string are different (that is, 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, that is, in the output specification. (QUEUE)

?(3) FILE WAS BEING MODIFIED-file.ext

Another user is modifying the file. (COMPIL)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?(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 you the job number to which the console was attached or from which the console is detaching. (ATTACH, DETACH)

?GIVING BACK TOO MUCH CORE

An internal problem in the DUMP program. Notify your system programmer or software specialist. (DUMP)

?HALT AT USER adr

Your 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)

%structure name HAS OTHER USERS-/REMOVE IGNORED

The specified structure is still being used by other jobs. The structure is removed from your search list but is not physically dismounted. (DISMOUNT)

?HAVE OPERATOR START OMOUNT, THEN TRY AGAIN

The command area where your requests are stored does not exist. (UMOUNT)



## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?HOST REJECTED CONNECTION

It is not possible to establish a connection with this system.  
(SET HOST)

### HOST SENT DISCONNECT

A system power failure usually produces this message in a network environment.

### ?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

While loading the bootstrap loader with the /switch, an unrecognizable block type was encountered by DTCOPY. (DTCOPY program)

### ?ILLEGAL BLOCK TYPE dev:file.ext

The block type used is not in the range 0-77. (MAKLIB)

### ?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 your program is illegal, such as dump mode for the terminal. (MAKLIB)

### ?ILLEGAL DENSITY FOR DRIVE

The density you specified was not compatible with those familiar to the monitor.

Acceptable arguments are: 200 bits/inch  
556 b/i  
800 b/i  
1600 b/i  
6250 b/i  
(SET DENSITY)

### ?ILLEGAL DEVICE - dev:

You specified a file structure name that is not a disk. (FILE)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?drive ILLEGAL DRIVE NAME

You requested a drive that is not recognized by the system, or you specified a drive that is in conflict with the unit or controller type required by the units of the file structure. (MOUNT)

### ? (0) ILLEGAL FILE NAME

A file name of zero was specified. (MAKLIB)

### ?ILLEGAL IN BATCH JOB

The ATTACH, DETACH, SEND, CCONT, SET WATCH, 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. (ATTACH)

### ?ILLEGAL NAME - name

You specified a structure name that does not exist. (MOUNT, DISMOUNT)

### ?ILLEGAL QUASAR MESSAGE FORMAT

The format of your request is invalid. Check the documentation for correct format. (ALLOCATE, DEALLOCATE, MOUNT, DISMOUNT)

### ?ILLEGAL QUEUE DEVICE

The queue name specified cannot be used with the given switch. (QUEUE)

### ?ILLEGAL QUEUE NAME xxx

The queue is not of the system queues.

### ?ILLEGAL SWITCH

A unrecognizable switch was used in the command string. (GLOB)

### ?ILLEGAL SYNTAX IN KJOB COMMAND

An illegal command syntax has been specified for a command to KJOB. (LOGOUT)

### ?ILLEGAL TO CREATE REQUEST FOR SOMEONE ELSE

Only the operator logged in under 1,2 can create queuing requests for other users. (QUEUE)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?drive name ILLEGAL UNIT

You specified a drive that is not a valid drive name. (MOUNT, DISMOUNT)

?ILLEGAL UO AT USER adr

An illegal UO was executed at user location adr.

?ILLEGAL WHEN EXECUTE ONLY

The program in memory is only executable. It cannot be written, examined, or changed.

?ILL INST. AT USER adr

An illegal operation code was encountered in your program.

?ILL MEM REF AT USER adr

An illegal memory reference was made by your 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. You should use the E command to first examine location adr-1 and then location adr to determine the illegal instruction. The index registers may also have to be examined.

?INCORRECT VALUE FOR /WRITE SWITCH:xxx

You specified the value xxx with the /WRITE switch. The only acceptable values are YES and NO. (MOUNT)

[structure IN: iiii OUT: ooooo USED; nnn]

The monitor has computed the disk usage on structure to be iiii input file quota, ooooo output file quota, and nnn blocks used. (LOGIN)

?INPUT (or OUTPUT) BLOCK TOO LARGE

A DECTape block number greater than 1101 (octal) was encountered. (DTCOPY program)

?INPUT (or OUTPUT) CHECKSUM OR PARITY ERROR

A read (or write) error has been detected. (DTCOPY program)

?INPUT DEVICE dev CANNOT DO OUTPUT AT USER adr

Output was attempted on a device that can only do input (for example, the card reader).

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?INPUT DEVICE DOES NOT EXIST XXX:

The system does not recognize the device name you specified (XXX:). (QUEUE)

### ?INPUT (or OUTPUT) DEVICE ERROR

The DECTape control unit has detected the loss of data or a missed block. (DTCOPY program)

### ?INPUT ERROR

An I/O error occurred while reading a temporary command file from the disk. The 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. (MAKLIB)

### ?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, DTCOPY encountered the end of file before it expected it. This may happen when copying a PDP-9 DECTape to a DECsystem-10 DECTape. (DTCOPY program)

### ?INSUFFICIENT CORE FOR QUEUE

There is not enough core in the 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. You are to retype your project-programmer number and password. (LOGIN)

### ?INVALID VOLUME SET NAME SPECIFIED

The volume set that you specified is not:

1. Mounted if you specified DISMOUNT, or
2. Allocated if you specified DEALLOCATE.

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?I/O TO UNASSIGNED CHANNEL AT USER adr

An attempt was made to do an OUTPUT, INPUT, OUT, or IN to a device that your program has not initialized.

?x IS AN ILLEGAL CHARACTER SWITCH

An illegal character or switch was encountered in the command string. (MAKLIB)

?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. The system is running the maximum number of jobs it can handle. Try to log in later. (LOGIN)
2. The system is running the maximum number of jobs, and the program attempted to create a new job. (SYSTAT, HELP, ATTACH)

?JOB HAS OUTSTANDING ETERNAL ENQ LOCKS SET

You cannot log out until you have removed the eternal lock for ENQUEUE/DEQUEUE. (KJOB)

?JOBNAME OR /SEQUENCE OR /REQUEST ID REQUIRED

You must specify the jobname when using the /KILL OR /MODIFY switch. (QUEUE)

?JOB NOT WAITING

The job specified is not waiting to be continued. (JCONT)

?JOBPEK UO REQUIRED, NOT IMPLEMENTED

DAEMON is not able to run because the necessary UO is not implemented by this installation. (DAEMON)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

JOB SAVED

The output is completed. (SAVE)

%JOBSTR UO FAILED - NO QUOTA ENFORCED

The JOBSTR monitor call was executed, and it failed; therefore, no quotas are in effect. (LOGOUT)

?KA10 FLOATING POINT INSTRUCTION AT USER PCxxx

Floating point capability is disabled. (SET FLOATING)

?LANGUAGE PROCESSOR CONFLICT

The use of the + construction has resulted in a mixture of source languages. (COMPIL)

?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)

?file structure LOCKED OUT BY OPERATOR

The operator has locked this structure, preventing any new accesses, in order that it may be removed. (MOUNT)

%LOG FILE DEVICE IS NOT DSK OR SPOOLED; WILL USE TTY

The log file device is not a disk or spooled; therefore, KJOB assumes TTY. (KJOB)

file structure name LOGGED OUT QUOTA n EXCEEDED BY m BLOCKS

Your allocation on the file structure named is greater than your logged out quota. You must go through the CONFIRM dialogue and delete files until you are under the quota allowed to log off. (KJOB, LOGOUT)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

%LOGICAL NAME WAS IN USE, DEVICE dev ASSIGNED

You previously assigned this logical name to another device. The logical name is cleared from the first device and assigned to the second.

LOGICAL NAME IN USE

You specified a logical name in your MOUNT command that is currently in use by the same job. MOUNT creates a unique logical name of the form "Mnnnnn" for this MOUNT request. (MOUNT)

?LOGIN PLEASE

You typed a command which requires that you be logged in. It cannot be accepted until you log in.

?LOGIN PLEASE TO USE SWITCH /CREATE

You 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 /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. (MAKLIB)

?file structure name LOOKUP FAILURE n

The LOOKUP to read the file failed; n is the disk error code.

LOOKUP FAILURE FOR AUXACC.SYS

The system was unable to find AUXACC.SYS (one of the system's master accounting files). The system is unable to determine quotas on your file structure. For MOUNT commands this means that no UFD can be created. (UMOUNT)

?LOOKUP FAILURE FOR INPUT FILE n file

DUMP cannot read the input file. (DUMP)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?LOOKUP FAILURE n ON DAEMON FILE

The LOOKUP to read the DAEMON file failed; n is the disk error code. (DAEMON)

%LOOKUP FOR HOME BLOCKS FAILED

The system is unable to locate or access HOME.SYS on the mounted structure. Your job will continue, this allowing unformatted structures to be accessed. (UMOUNT)

?MAKLIB SYNTAX ERROR

An illegal command string was entered; for example, the equal sign was omitted or a program name was specified for the output file. (MAKLIB)

?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 master file directory. (LOGIN)

?MAY NOT LOGIN { LOCAL  
REMOTE  
DATA SET  
BATCH JOB SUBJOB  
REMOTE CTY  
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 your job's search list is assigned a logical name, and only physical device names are recognized. You 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 your 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 to record the pertinent information about the error for field service personnel.

You must start a new copy of your program by typing the appropriate monitor command R, RUN, or GET. You should not start the program over by typing START, because the error is likely to reoccur or the program operate with incorrect data.



## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?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)

### ?MOUNT INCOMPLETE

UMOUNT was unable to complete your MOUNT request. Other messages give the reasons. (UMOUNT)

### ?MOUNT MUST BE INVOKED AS A COMMAND

The FILE command controls DECTapes. You must use the MOUNT command to mount the DECTape before you can access it. (FILE)

### ?MOUNT UNSUCCESSFUL

Was unable to process the MOUNT request. Other messages accompany this one to give the reasons. (UMOUNT)

### ?MUST BE IN OWNER'S PROJECT TO REQUEST SINGLE ACCESS

You may not request single-access (/SINGLE switch) unless you have the same project number as the owner of the file structure. This requirement is enforced because a user with single access may execute super-USETI/USETO UUOs. (UMOUNT)

### ?MUST BE LOGGED IN

You must be logged in to queue MOUNT, DISMOUNT, and FILE requests. (Note that with a change to the monitor, these commands may be used with the check or HELP functions without you being logged in.) (UMOUNT)

### ?MUST HAVE "EXE" EXTENSION

To use the RUN command, the specified file must have a file extension of .EXE. Save the source program with the SAVE or NSAVE command. (SAVE)

### ?NEED 5.03 OR LATER FOR REATTACH COMMAND

The REATTA program depends on UUOs available in the 5.03 release of the monitor. You attempted to run the program using an older monitor. (REATTA)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?NESTING TOO DEEP

The @ construction exceeds a depth of six. This may be due to a loop of indirect command files (COMPIL).

### NO COMMANDS IN QUEUE

There are no requests (of the specified type; MOUNT, DISMOUNT, or FILE) in the queue from any of the users. This message may be issued in response to any one of the following commands: MOUNT /CHECK, DISMOUNT /CHECK, or FILE C. (FILE, MOUNT, DISMOUNT)

### ?NO CORE ASSIGNED

No core was allocated when the GET command was given and no core argument was specified in the GET.

### ?NODE DOES NOT SUPPORT REMOTE TERMINALS

The node you specified does not have the facilities to service your terminal. (LOCATE, SET HOST)

### %NODE(x) LOCATED BUT OFF-LINE

The node you specified is not on the network at present. (LOCATE)

### ?NODE name NOT IN NETWORK

The node you specified as name is not in the network. (NETWORK)

### %NO DEVICE FOR DIRECTORY

A FILE command has been issued and you have no device in your active search list that is eligible to have the directory written on it. That is, the search list does not contain a structure that has both a UFD for you and a status of /CREATE and /WRITE. The command is processed but no directory is created. (When writes directories, it attempts to put them on the first structure in your search list that has both a UFD and a status of /CREATE and /WRITE.) (FILE)

### 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). (DTCOPY program)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?NO ENTRY IN AUXACC.SYS  
NO SEARCH LIST OR UFDS CREATED

If you have no entry in AUXACC.SYS, LOGIN does not create UFDS or a search list. You are logged-in and have UFDS if they existed previously. You may write only on file structures that have UFDS or read all file structures. You may also create a file structure search list with SETSRC. You can create UFDS on those file structures for which you have an entry in QUOTA.SYS by using the MOUNT command. (LOGIN)

NO ENTRY IN QUOTA.SYS OR AUXACC.SYS

You do not have a UFD on the specified structure. Your ability to access and create files on the structure depends on the individual protection of the structure's other UFDS and files. (UMOUNT)

%NO FILE NAMED file specification

The file specification you typed was incorrect. (DELETE)

%NO HELP AVAILABLE

The file HLP:PLEASE.HLP could not be found. (PLEASE)

?NO INFORMATION IS CURRENTLY AVAILABLE ON "name"

The feature you specified is not listed in the HLP: structure. There is no available information about name. (HELP)

%NO INFO ON name

You 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 FOR YOUR JOB

You have specified the WAIT or CHECK option in your FILE, MOUNT, or DISMOUNT command and there are no pending requests in the queue. (UMOUNT)

?NO NETWORK OR DN60 SOFTWARE

The command you typed initiates network software, which is not implemented on the system you are using.

?NON-EXISTENT DRIVE dev:

You have specified a drive that does not exist in the system. (UMOUNT)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

%NON-EXISTENT FILE file name

The file specified for input could not be found. (DIRECT, QUEUE)

?NON-EX MEM AT USER adr

Usually due to an error in the monitor.

?NO NODES MEET CONSTRAINTS

The switch(es) that you specified in the command string eliminated all nodes. (NETWORK)

?NO OPERATOR IN ATTENDANCE

There is no operator available. However, your message will be sent. (PLEASE)

?NO OPR JOB FOR THIS REQUEST

An operator request has been issued, but there is no OMOUNT running and enabled to service the request. Your request is still queued unless the /PAUSE switch was given. (UMOUNT)

?NO PRIVILEGES TO SET CPU

You do not have the privilege set by LOGIN from ACCT.SYS to change the CPU specification. You should request that these privilege bits be set by the system manager. (SET CPU)

?NO PRIVS TO SET PRIORITY THAT HIGH

You do not have privileges to set a high priority. (SET HPQ)

?NO PRIVS TO UNSPOOL

You do 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. (MAKLIB)

?NO QUEUE SPECIFIED IN /KILL OR /MODIFY

You must specify a queue when using these switches. (QUEUE)

?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)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?NO ROOM IN QUEUE, TRY AGAIN LATER

The [3,3] command area (where your requests are stored) is full.  
(FILE, MOUNT, DISMOUNT)

?(14) NO ROOM OR QUOTA EXCEEDED - file.ext

There is no room on the file structure or your 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, you receive this message every minute until the CPU is started or you type a new SET CPU command.

?NO START ADR

Starting address or reenter address is zero, because you 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 STRUCTURE NAME

You did not specify a device in your DISMOUNT command. (UMOUNT)

?NO SUCH DEVICE

The device name does not exist or was not assigned to this job.  
(WHERE, ASSIGN)

%NO SUCH FILES AS file spec

A file has been specified that does not exist in your default directory.

?NO SUCH JOB

An attempt was made to attach to a job that has not been initialized. (ATTACH)

?NO SUCH STR

A nonexistent file structure was specified. (KJOB)

?NO SUCH TTY

You specified a line number that does not exist. Try using the job number format. (SEND)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?NO SUCH UNIT

The unit does not exist or all units of this type are in use.  
(ASSIGN)

?NO SYS

This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. (UMOUNT)

?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. Use the SAVE command to create a core image (.EXE) file. This can be run using the RUN command.

?NOT A SPOOLABLE 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)

?NOT AVAILABLE

The operator has restricted use of the specified feature, or the console switches are set to disallow the use of a programmable address break.

?drive NOT AVAILABLE

The drive you specified is not currently available. (UMOUNT)

?dev NOT AVAILABLE

The specified device does not exist or is assigned to another user. (MAKLIB)

?command NOT CODED

A command that is not in this version of DUMP was specified in the command string. (DUMP)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?NOT ENOUGH ARGUMENTS

An insufficient number of files of one type has been specified.  
(MAKLIB & REASSIGN)

### ?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. (UMOUNT)

### 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 missed because of  
the incorrect sequence in the command string. (MAKLIB)

### ?file.SAV NOT FOUND

The program file requested cannot be found on the system device  
or the specified device. If you have not created a core image  
file (.EXE or .SAV), the system cannot run the program. Use the  
SAVE command to create a core image file before using RUN to  
execute the program.

### unit-id NOT MOUNTED

The file structure is mounted but the specified unit is not.  
According to STRLST.SYS, the unit should belong to the structure.  
Consult the system administrator. This message is followed by  
the message "REQUEST QUEUED TO READY DRIVES." (UMOUNT)

### unit-id NOT READY

The requested file structure exists but the unit is not ready.  
This message is followed by the message "REQUEST QUEUED TO READY  
DRIVES." (UMOUNT)

### ?NOT YET SUPPORTED COMMAND CODE switch

A switch has been specified that is not implemented. (QUEUE)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

NO UFD CREATED

You cannot have a UFD on this structure. Your ability to access and create files on the structure depends on the individual protection of the structure's other UFDs and files. You may access the file structure, but you cannot write in your disk area until you have a UFD. (UMOUNT)

?NULL DEVICE ILLEGAL

A colon has been found without a preceding device name. (QUEUE)

?NULL SFD ILLEGAL

An empty SFD has been created. Use the ZERO command to delete the SFD, then delete the .SFD file. Create a new SFD, if necessary, then copy or write files into the new SFD.

?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 executed. You 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. (DTCOPY 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)

?OPEN UO FAILED FOR structure

The specified structure tried to execute an OPEN monitor call, but it failed. (LOGOUT)

[OPERATOR AT node HAS BEEN NOTIFIED AT time]

The operator at the specified node has received your message at the time. (PLEASE)



## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### OPERATOR NOTIFIED

A request is queued to the operator to perform a specified action. (UMOUNT)

### OPERATOR NOT ON DUTY

SCHED 400 is set (that is, there is no operator on duty to handle MOUNT and DISMOUNT requests). If you did not specify the /SHOVE switch in your request, then the MOUNT or DISMOUNT cannot be completed. If you did specify the /SHOVE switch then you receive the message, "SHOVING REQUEST ANYWAY..." and normal processing resumes. (UMOUNT)

### OPERATOR NOT ON DUTY - /REMOVE IGNORED

The specified structure will be removed from your search list but will not be physically removed from the drive. (UMOUNT)

### OPER ONLY COMMAND

You issued an operator restricted command. (OPSER)

### OTHER USER - CANNOT SINGLE ACCESS

You issued a MOUNT command with the /SINGLE switch specified and others have use of the drive. The MOUNT will proceed and the /SINGLE switch will be ignored. (UMOUNT)

### 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 your core area, or the high segment is write-protected and you do 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 (for example, the line printer).

### ?OUTPUT DEVICE ERROR

An error has occurred on the output device. (FILCOM)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?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)

### ?OUTPUT (or INPUT) PREMATURE END OF FILE

When copying a DEctape, DTCOPY encountered the end of file before it expected it. This may happen when copying a PDP-9 DEctape to a DECSYSTEM-10 DEctape. (DTCOPY program)

### PATH TO HOST SYSTEM WAS LOST

For any number of reasons, the connection to the host system was not completed. Use CTRL/C to reconnect. (SET HOST)

### ?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 you already had a job initialized at that terminal. (LOGIN)

### ?PLEASE LOGIN AS [OPR]

The operator is the only person who can initialize DAEMON by typing R DAEMON.

### ?PLEASE START OVER

The LOGIN program ran over its time limit. Your temporary job is killed. You must start LOGIN again. (LOGIN)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?PLEASE TYPE ^C FIRST

To CONTINUE the program, after having used CSTART or CCONTINUE, the program must be interrupted with a CTRL/C.

?PPN HAS EXPIRED

The current date is greater than the expiration date of the project-programmer number. You may not log in until expiration date is changed by the system manager. (LOGIN)

?PPN OR DEVICE PERMITTED ONLY ON FILE F OR FILE Z CMDS.

You specified a project-programmer number or a device with an implied project-programmer number (for example, NEW:) in a FILE command that does not permit specification of a project-programmer number. A project-programmer number may be specified only on the FILE F and FILE Z commands. (FILE)

?PROGRAM ERROR WHILE RESETTING MASTER DEVICE

FUDGE2 cannot find the master device or cannot find the program on the master device. (FUDGE2)

?PROGRAM NOT PRIVILEGED - IGNORED!

UMOUNT must be run from SYS: or NEW: or you must be [1,2] to queue requests. (UMOUNT)

?PROJECT 1 MAY NOT BE PTY

Project 1 is never allowed to login over a pseudo-TTY. (LOGIN)

?PROJECT-PROGRAMMER NUMBER ILLEGAL WITH A SESSION COMMAND

The SESSION command does not accept PPNs. It is used to change ACCOUNT strings and REMARKS only. (SESSION)

?PROTECTION FAILURE file-specification

You do not have privileges to access the indicated file (file specification). (FILE)

?(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. (DTCOPY program)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?QUENFI NO FILES IN REQUEST

An illegal queue request was made; the monitor aborted the request.

QUOTA.SYS LOOKUP FAILURE

The LOOKUP to read QUOTA.SYS failed. (UMOUNT)

QUOTA.SYS NOT ON STRUCTURE

QUOTA.SYS is not part of this structure. You may still use the file structure, but no UFD will be created. (UMOUNT)

QUOTA.SYS READ ERROR

An I/O error occurred while reading QUOTA.SYS. (UMOUNT)

QUOTA.SYS OR AUXACC.SYS READ ERROR

An I/O error occurred while QUOTA.SYS or AUXACC.SYS was being read. For MOUNT commands this means that no UFD can be created. (UMOUNT)

QUOTA.SYS WRONG FORMAT VERSION

The wrong version of QUOTA.SYS is on the file structure being mounted. Consult the operator. (UMOUNT)

%REASSIGN ILLEGAL FOR RESTRICTED DEVICE - /RELEASE ASSUMED

You may not reassign a restricted device. If you attempt to do so, the device is released from your job and returned to the pool of restricted devices. (UMOUNT)

%REASSIGN JOB # NOT ASSIGNED - /RELEASE ASSUMED

You attempted to reassign a device to a job that doesn't exist. The device is released from your job. (UMOUNT)

[RECOMPUTING DISK USAGE]

The monitor is recomputing the disk usage on the default directory, before the user can successfully LOGIN.

[RECOMPUTING DISK USAGE ON structure]

Disk usage is being recomputed on the specified structure. (LOGOUT)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?/REELID REQUIRES MAGTAPE

You specified a /REELID switch but the requested device is not a magnetic tape. The /REELID switch may be specified only when mounting magnetic tapes. (UMOUNT)

### ?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 FILE NAME ALREADY EXISTS - file.ext

The new file name on a RENAME command already exists. (COMPIL)

### REQUEST QUEUED

Your request to perform a specified action has been queued to the operator. (UMOUNT)

### REQUEST QUEUED TO MOUNT UNITS

A request is queued to the operator to mount and ready the packs on the proper drives. (UMOUNT)

### REQUEST QUEUED TO READY DRIVES

The requested structure is mounted but one or more drives (as specified by previous messages) are not ready. A request is queued to the operator. (UMOUNT)

### ?REQUIRES DEVICE NAME

The device name or file structure name is required with the MOUNT and DISMOUNT commands. (UMOUNT)

### ?REQUIRES /REELID SWITCH

You must include the /REELID switch and its argument in the command string when you want to have a magnetic tape mounted. (MOUNT for magnetic tapes) (UMOUNT)

### ?REQUIRES /VID SWITCH

You must include the /VID switch and its argument in your MOUNT command for DECTapes and for all input-only devices. (UMOUNT)

### ?RIGHT BRACKET REQUIRED IN DIRECTORY

The project-programmer number must be enclosed in square brackets. (QUEUE)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

%SEARCH LIST DOES NOT ALLOW CREATES

There are no file structures available to you on which you can write. Run MOUNT or SETSRC to modify your search list as necessary. (LOGIN)

%SEARCH LIST IS EMPTY

There are no file structures in the DSK: search list that are available to you. You can run the SETSRC program to modify your search list. (LOGIN)

SHOVING REQUEST ANYWAY...

SCHED 400 is set (that is, there is no operator on duty to handle MOUNT and DISMOUNT requests). However, you have specified the /SHOVE switch so the request will be queued even though there may be no one at the computer to MOUNT or DISMOUNT the device(s). (FILE, MOUNT, DISMOUNT)

?SINGLE ACCESS BY JOB n

You requested to have a structure mounted that is already mounted by job n with the /SINGLE switch specified. (UMOUNT)

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

You are not scheduled to LOGIN at this time. You should try again when you are allowed to login. (LOGIN)

?SORRY, CAN'T OPEN DSK, PLEASE CALL THE OPERATOR

This message is returned from the GRIPE program.

?SORRY, CAN'T 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.

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?SPECIFIED DEVICE IS NOT YOURS

The resource that you attempted to DISMOUNT is not exclusively assigned to you. Other users are accessing the resource.  
(DISMOUNT)

%SPECIFIED LOGICAL NAME WAS IN USE

You specified a logical name in your MOUNT command that was already in use by the same job. The device with which the logical name was previously associated is no longer associated with the logical name. The logical name is now associated with the newly mounted device. (UMOUNT)

?STATION NUMBER INVALID

The requested station number is not recognized by the system.  
(LOCATE)

STR HAS ONLY n UNITS

You have requested more drives than are necessary to accommodate the structure. The MOUNT proceeds with only the number of drives necessary. (UMOUNT)

STRUCTURE ALREADY MOUNTED

[MOUNT COUNT = n]

The requested file structure already exists and does not need to be physically mounted. "n" is the number of users (not including yourself) who currently have the structure mounted. (UMOUNT)

?STRUCTURE NOT MOUNTED

You requested that a structure be dismantled. The specified structure either is not a valid name or it is not mounted.  
(UMOUNT)

%STRUCTURE UFD LOOKUP FAILURE nn

A LOOKUP was executed on the specified structure, and it failed.

?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)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

%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)

?switch SWITCH ILLEGAL

The switch specified cannot be used with the given queue name. (QUEUE)

?SWITCH SYNTAX ERROR

The switch you used was invalid, or you typed the slash with no switch name. (PLEASE)

?SWITCH VALUE REQUIRED ON SINCE

A /SINCE was specified with no argument.

?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 UO gave an error return. Notify the operator. (KJOB, LOGOUT)

SYSTEM ERROR AT LOC nnn

This is a system error and therefore is not a direct fault of the user. It is printed for potential diagnostic use. (UMOUNT)

?file structure name SYSTEM ERROR - DSKCHR UO FAILED

This is a system error and therefore is not a direct fault of the user. It is printed for potential diagnostic use. (UMOUNT)

?file-structure name SYSTEM ERROR - JOBSTR UO FAILED

This is a system error and therefore is not a direct fault of the user. It is printed for potential diagnostic use. (UMOUNT)



## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?SYSTEM ERROR - SYSPHY UOO FAILED

This is a system error and therefore is not a direct fault of the user. It is printed for potential diagnostic use. (UMOUNT)

### ?SYSTEM ERROR - WRONG FORMAT VERSION NUMBER IN STRLST.SYS

STRLST.SYS is not in a format that is understood by MOUNT. This is a system error and therefore is not a direct fault of the user. It is printed for potential diagnostic use. (UMOUNT)

### ?SYSTEM I/O ERROR AT loc STATUS = code

An unexpected I/O error occurred at location "loc". "code" is the status code returned by the GETSTS UOO. This is a system error and therefore is not a direct fault of the user. It is printed for potential diagnostic use. Consult the system administrator. (UMOUNT)

### ?SYSTEM NOT AVAILABLE

The operator has used the SET SCHED command to prevent LOGINS from timesharing terminals. The message of the day is still printed. (LOGIN)

### ?TABLE OVERFLOW - CORE UOO FAILED TRYING TO EXPAND TO xxx

The GLOB program requested additional core from the monitor, but none was available. (GLOB)

### ?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 UOO has been issued to stop timesharing on the system at the indicated time.

### ?TOO FEW ARGUMENTS

A command has been printed, but necessary arguments are missing.

### ?TOO MANY FILE NAMES OR PROGRAM NAMES

More than 40 program names or file names were specified in the command string. You should separate the job into several segments. (MAKLIB)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?TOO MANY FILES, LAST PROCEED IS file.ext

You requested (either explicitly or implicitly) more than 22 files to be filed onto or recalled from DECTape. The request is processed up to the point where "file.ext" is transferred. (FILE)

?TOO MANY FILE STRUCTURES

The number of file structures exceeds the capacity of the monitor data base. The current limit is 36 (decimal). (ONCE)

?TOO MANY NAMES or ?TOO MANY SWITCHES

Command string complexity exceeds table space in the COMPIL program. (COMPIL)

?TRANSMISSION ERROR ON INPUT DEVICE dev

A transmission error has occurred while reading data from the specified device. (MAKLIB)

?TTYnn ALREADY ATTACHED

You tried to attach your terminal to a job, but your terminal was already attached to the job. (ATTACH)

?TYPE SYS H FOR HELP

You used an invalid format for the command. The argument H will print a help file on your terminal.

%structure UFD RENAME FAILURE nn

A RENAME was executed on the specified structure, and it failed. (LOGOUT)

?UNABLE TO COMPLETE SET HOST COMMAND

The instruction failed. Wait a few minutes, then try to SET HOST again. (SET HOST)

?UNDEFINED NETWORK NODE

The system does not recognize the node name you specified. (SET HOST)

?UNEQUAL AMOUNT OF MASTER AND TRANSACTION PROGRAMS

On a replace request, the number of master programs (or files) does not equal the number of transaction program (or files). (MAKLIB)

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### ?UNKNOWN COMMAND

The monitor passed a command to COMPIL that COMPIL does not recognize. (COMPIL)

### ?UNKNOWN DEFAULT FOR SWITCH switch

The default condition is not known for the specified switch. (DUMP, QUEUE)

### ?UNKNOWN DEVICE OR STRUCTURE NAME

The device or 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)

### ?UNKNOWN SWITCH switch

The switch named has been typed incorrectly. (DUMP, QUEUE)

### ?UNKNOWN SWITCH VALUE n

The argument specified with the switch has been misprinted. (DUMP, QUEUE)

### ?UNMATCHED OPEN PARENTHESIS

You must use the close parens to terminate a string that was begun by open parens. (DIRECT)

### ?UNRECOGNIZABLE SWITCH

An ambiguous or undefined word followed a slash. (COMPIL)

### ?UNRECOGNIZED NAME IN DATE/TIME

An illegal character appears after a time/date - type switch. (for example, /SINCE:L-APR-L971.)

### %UNRECOGNIZED SWITCH switch - IGNORED

An illegal (unrecognizable) switch has been specified for KJOB, it is ignored. (LOGOUT)

### ?UOO ERROR AT USER adr

This message accompanies many error messages and indicates the location of the UOO that was the last instruction the user program executed before the error occurred.

## SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

### n VERIFICATION ERRORS

On a word by word comparison requested with the /V switch, n discrepancies have been detected between the input DECTape and output DECTape. (DTCOPY program)

### WAITING...2 ^C's TO EXIT

A request has been queued to the operator and the command is waiting for the operator to complete the request. If you do not want to wait for completion of the operator's action, you can type two CTRL/Cs 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. (Note that this message is not issued when the /NOWAIT switch is specified.) (UMOUNT)

### ?WAITING FOR A VALID PASSWORD

The LOGIN program waits for two minutes after printing the password prompt on your terminal. If you don't type your password within two minutes, LOGIN prints this message on your terminal. (LOGIN)

### WAITING FOR CONNECT CONFIRM

The new host has not yet recognized your terminal. (SET HOST)

### [WAITING FOR structure UFD INTERLOCK]

LOGOUT attempts to set the monitor UFD interlock before doing quota enforcement. If another job has it interlocked already, LOGOUT will wait. (LOGOUT)

### 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. (LOG)

### %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

You have 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. (MAKLIB)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

?dev WASN'T ASSIGNED

The device is not currently assigned to your job and cannot be deassigned or reassigned by the job.

?WASN'T DETACHED

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)

?WILD DEVICES ILLEGAL

You cannot use the wildcard construction in the device specification. (DELETE)

?WRITE LOCK ERROR

An attempt was made to write on a write-locked DECTape. (DTCOPY 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 REACT to change the accounting files. (LOGIN)

drive WRONG UNIT TYPE

The user has requested a drive which is inappropriate for the pack to be mounted. (For example, an RP03 pack cannot be mounted on an RP04 drive.) (UMOUNT)

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

YOU ARE LOGGED IN AS n,m

When you log in with a unique programmer number (project, #), this message informs you of the project-programmer number that LOGIN assigned. (LOGIN)

?YOU DON'T HAVE PRIVILEGES TO WRITE { DAEMON } FILE  
  { CCL     }

You attempted to write in a file to which you did not have access. (DAEMON)

?YOUR STATION DOES NOT SUPPORT SET HOST

You will not be able to reestablish yourself with another system from the system you are currently using. (SET HOST)

APPENDIX A  
FUNCTIONAL GROUPS OF COMMANDS

Table A-1 lists the operating system commands, divided into groups by function. Next to the name of the group is a list of the commands in that group. The commands in each group are arranged in alphabetical order. Each command is accompanied by a short description of its function.

For an explanation of each functional group, read Sections 2.1 through 2.11.

NOTE

Table A-1 lists the commands and programs that are described in Section 2.12. Do not use the commands and programs listed in Table A-1 until you are familiar with them.

FUNCTIONAL GROUPS OF COMMANDS

Table A-1  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
Job-Control	ATTACH	Connects your terminal to a previously DETACHED job.
	DETACH	Releases your terminal from the current job.
	DISABLE	Disables the privileges that you previously ENABLED.
	ENABLE	Enables privileges that are assigned to your job.
	KJOB	Terminates your job.
	LOGIN	Allows you to access the system.
	OPSER (program)	Allows you to create subjobs on your terminal.
REATTACH (program)	Transfers your job from one terminal to another.	
Information	CORE	Prints statistics about your job's core area.
	CPUNCH	Prints a list of the jobs in the card-punch queue.
	DAYTIME	Prints the date and time.
	DIRECTORY	Prints a list of all the files in your directory.
	DSK	Prints the disk-usage information for your job.
	HELP	Prints a message which shows the formats of the HELP command.
	INITIA	Prints information about the system, your job, and your terminal.



FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
Information (Cont.)	LOCATE	Prints the node to which your terminal is logically connected.
	NETWORK	Prints a list of the nodes in the network to which your terminal is logically connected.
	NODE	Prints information about the node to which your terminal is connected.
	PJOB	Prints information about the specified job.
	PLOT	Prints a list of the jobs in the plotter queue.
	PRINT	Prints a list of the jobs in the line-printer queue.
	QUEUE	Prints a list of all the jobs in all the queues.
	QUOLST (program)	Prints the disk-usage information for your directory.
	RESOURCES	Prints a list of all available devices and file structures.
	SCHEDULE	Prints the system status for timesharing and operator coverage.
	SET WATCH	Sets the system to automatically print incremental job statistics.
	SYSTAT	Prints all available information about the system; in particular, every job on the system.
	TIME	Prints the runtime since you logged-in and since the last TIME command.

FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
Information (Cont.)	TPUNCH	Prints a list of all the jobs in the paper-tape-punch queue.
	CTRL/T (USESTAT)	Prints the status of your job.
	VERSION	Prints the version number of the program currently in memory.
	WHERE	Prints the node to which the specified device is connected.
Terminal-Control	INITIA (program)	Sets and reports on the parameters of your terminal.
	SET TERMINAL TERMINAL SET TTY TTY	Sets the specified parameters for your terminal.
Terminal Communication	PLEASE	Begins communication with the computer operator.
	SEND	Sends a message to a specified terminal.
File-Handling	BACKUP (program)	Saves and restores copies of disk files on magnetic tape.
	CLOSE	Closes the open files on a device.
	COPY	Copies files on the same device or from device to device.
	CPUNCH	Enters files in the card-punch queue.
	CREDIR (program)	Creates directories and sub-file directories.
	DELETE	Removes the specified file from your directory.

FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
File-Handling (Cont.)	DIRECTORY	Prints a list of the files in your directory.
	DUMP	Produces printable ASCII output from a core-image file.
	EOF	Writes an end-of-file mark on magnetic tape.
	FILE	Transfers files from disk to DECTape and DECTape to disk.
	LIST	Lists a file on the line printer.
	MAKE	Creates a file with the text editor TECO.
	PLOT	Enters files in the plotter queue.
	PRESERVE	Changes the protection code of the specified file.
	PRINT	Enters files in the line-printer queue.
	PROTECT	Changes the protection code of the specified file.
	QUEUE	Enters files in the specified queue.
	RENAME	Changes the name of a file.
	SET DEFAULT PROTECTION	Changes your default protection code.
	SETSRC (program)	Allows you to change your job's search list.
SUBMIT	Enters files in the batch input queue.	

FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
File-Handling (Cont.)	TECO	Calls the TECO text editor to edit an old file.
	TPUNCH	Enters files in the paper tape punch queue.
	TYPE	Prints the contents of the specified file on your terminal.
	ZERO	Eliminates the specified directory from the specified device.
Device-Handling	ASSIGN	Allocates an I/O device to your job and assigns logical names to the devices.
	BACKSPACE	Spaces a magnetic tape backward a specified number of files or records.
	CLOSE	Closes the open files on a device.
	CPUNCH	Enters files in the card-punch queue.
	DEASSIGN	Releases any devices from your job and clears logical names.
	DISMOUNT	Releases disks and tapes from your job, and removes file structures from your search list.
	EOF	Writes an end-of-file mark on magnetic tape.
	FINISH	Closes any open files and releases a device from your job, clearing logical names.
	LABEL	Writes an identifier onto DEctape.
LIST	Lists a file on the line-printer.	

FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
Device-Handling (Cont.)	MOUNT	Requests the operator to mount a device, assign it to you, and adds the file structure to your search list, if necessary.
	PRINT	Enters files in the line-printer queue.
	QUEUE	Enters files in the specified queue.
	REASSIGN	Passes a device from one job to another.
	REWIND	Rewinds a magnetic tape or DECTape.
	SET BLOCKSIZE	Sets the default blocksize for a magnetic tape.
	SET CDR	Sets the file name of the next file to be read by the card reader.
	SET DENSITY	Sets the default density of the magnetic tape.
	SET RETRY	Controls error-reporting from magnetic tapes.
	SET SPOOL	Changes the list of devices that will be spooled for your job.
	SKIP	Moves a magnetic tape forward the specified number of files or records.
	TPUNCH	Enters files in the paper tape punch queue.
UNLOAD	Rewinds and unloads a magnetic tape, DECTape, or disk.	

FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
Program- Preparation	CLOSE	Closes the open files on a device.
	COMPILE	Translates a source program, using the appropriate compiler.
	CREF	Generates a cross-reference listing and sends it to the line printer.
	D(eposit)	Puts data into your memory area.
	DCORE	Writes a core-image file of your program, including accumulators and job tables.
	DDT	Loads VMDDT, if necessary, and runs the debugger.
	DEBUG	Loads a program and a debugger, then starts the debugger.
	E(xamine)	Displays the contents of the specified core address.
	EOF	Writes an end-of-file mark on magnetic tape.
	FILCOM (program)	Compares two files and outputs the differences between them.
	FUDGE	Creates a library .REL file from a temporary file.
	GLOB (program)	Produces a cross-referenced listing of global symbols.
	LOAD	Compiles a source file, and loads .REL files into memory.
	MAKE	Creates a file with the text editor TECO.
MERGE	Loads a second program into memory, to be run with the first.	

FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
Program-Preparation (Cont.)	SET BREAK	Establishes an interrupt point at a specified location in your program.
	TECO	Calls the TECO text editor to edit an old file.
Program-Control	CTRL/C (HALT)	Stops execution of a program, leaving your terminal at monitor level.
	CCONTINUE	Continues program execution, leaving your terminal at monitor level.
	CONTINUE	Continues program execution, leaving your terminal at user level.
	CORE	Changes the amount of memory allocated to your job.
	CSTART	Starts program execution, leaving your terminal at monitor level.
	EXECUTE	Compiles, loads, and runs a program.
	GET	Loads a program into memory.
	JCONTINUE	Continues execution of a program after a device error.
	MERGE	Loads a second program into memory, to be run with the first.
	NSAVE NSSAVE	Writes .EXE files on the specified device.
	OSAVE OSSAVE	Writes the core image of your program on the specified device.

FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
Program-Control (Cont.)	R	Runs a system program.
	REENTER	Restarts program execution at a predefined entry point.
	RUN	Begins execution of a program.
	SAVE SSAVE	Writes a file of your core image on the specified device.
	SET CPU	Changes the processor on which your job runs.
	SET DEFAULT BUFFERS	Allows you to change the number of memory buffers.
	SET DSKFUL	Halts your job without destroying your core image when disk quota is exceeded.
	SET DSKPRI	Allows you to set the priority of your disk transfers.
	SET FLOATING POINT	Sets the system to simulate a floating point.
	SET HPQ	Sets the priority of your job in the run queue.
	SET PHYSICAL	Changes the amount of physical core you can use.
	SET TIME	Sets a time limit for your job.
	SET VIRTUAL LIMIT	Sets the current virtual page limit.
SETSRC (program)	Allows you to change your job's search list.	
START	Begins execution of a loaded program.	



FUNCTIONAL GROUPS OF COMMANDS

Table A-1 (Cont.)  
Functional Groups of Commands/Programs

Functional Group	Command/Program	Functional Description
Network	ASSIGN	Allows you to access a device on another system in the network.
	NETWORK	Prints a list of the nodes in the network.
	NODE	Prints information about the configuration of the network.
	SET HOST	Connects your terminal to the specified host system.
	WHERE	Prints the node to which the specified device is connected.



## APPENDIX B

### SWITCH.INI FILES

You can create a file in your disk area into which you can put switches for programs that read the SWITCH.INI file.

BACKUP	CREDIR
DIRECT	FORTRAN
INITIA	LINK
LOGIN	MAKLIB
QUEUE	RUNOFF
SORT	

The SWITCH.INI file allows you to automatically override the system defaults of these programs. The SWITCH.INI file must reside in your directory area. A SWITCH.INI file can contain two types of command lines.

The first type of line is written in the following format:

```
program-name/switch/switch/.../switch
```

Where: **program-name** is a program name such as DIRECT or LOGIN or a command name which runs one of the programs, such as PRINT.

**/switch** is a valid switch for the named program.

#### Example

```
DIRECT/DETAIL/NOSUMMARY
```

When you run the program, the switches in SWITCH.INI will be used as the defaults instead of any program-defined defaults. When you run a program, the system searches your directory for a file called SWITCH.INI. If the file is not found the program uses any program defaults. If the system finds the file but does not find a line for the program, it uses any program defaults. When the system finds your SWITCH.INI file and the line for the program, the program uses the switch values that you have specified in your SWITCH.INI file instead of any program default values.

You can override any switch in your SWITCH.INI file by issuing a command string to the specified program containing a complement of the switch in your SWITCH.INI file.

For example, you could have a SWITCH.INI file in your area that contains the following:

```
DIRECT/FAST
```

## SWITCH.INI FILES

Then, when you issue the following command:

```
.DIRECT
```

The system will print a fast (shortened) listing of your directory area. You could override the SWITCH.INI file by typing other switches after the command. For example:

```
.DIRECT/NORMAL
```

The second type of line that can appear in your SWITCH.INI file is written in the following format:

```
program-name:option-name/switch/switch.../switch
```

Where: **program-name** is a program name such as DIRECT, LINK, LOGIN, INITIA, or QUEUE.

**option-name** is the same as used for the /OPTION switch. (Refer to the OPTION switch in the command descriptions in Chapter 2.)

You use the second type of line to override both program defaults and any defaults that you might have previously specified in the SWITCH.INI file. You refer to this type of line in SWITCH.INI by including the /OPTION: switch in a command string to a program. If you specify an option name in the command string that does not appear in your SWITCH.INI file, the program prints a warning message and uses the program defaults.

Assume that you create a SWITCH.INI file that contains:

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

If you then type the DIRECT command, the program will print a fast listing showing both the actual unit names (instead of the structure names) and the summary line on your terminal. When you want the program to print a normal directory listing, you must type a command string to DIRECT that includes the /NORMAL switch. Note that disk unit names, not structure names, and the summary line will still be printed.

You can type the following command string to automatically list the length of the files in words, instead of blocks, and to update the access date of files with 25 blocks or less:

```
DIRECT/OPTION:THISRUN
```

### Example

The following example shows how to use a SWITCH.INI file.

First, log in to the system:

```
.LOG 27,5055(RET)  
JOB 49 RZ024A KL #1026/1042 TTY31  
PASSWORD:(RET)  
10:27 05-DEC-79 WED
```

## SWITCH.INI FILES

The following INITIA command shows your terminal characteristics. Since you haven't used the SET TTY TYPE command, and you do not have a SWITCH.INI file to set the terminal type, the system defaults to terminal type TTY.

.INI TTY

```
RZ024A KL#1026/1042 10:27:27 TTY31 SYSTEM 1026/1042
CONNECTED TO NODE KUL26 (26) LINE #31
JOB 49 USER MCCARRON, M [27,5055]
DSKC: KL1026 SYSTEM DISK DSKC
DSKB: KL1026 SYSTEM DISK DSKB
```

TYPE:TTY	NOLC	NOTABS	NOFORM
ECHO	CRLF	WIDTH:72	NOGAG
NODISPLA	FILL:1	NOTAPE	PAGE
BLANKS	ALTMODE	RCVSPEED:2400	XMTSPEED:2400
NODEBREA	NOTIDY	NORTCOMP	NOSLAVE
NOHDX	REMOTE		

Create a file named SWITCH.INI, with a command to LOGIN that sets your terminal type.

.SOS SWITCH.INI

INPUT: SWITCH.INI

00100 LOGIN/TERM:VT52

00200

\*ES

Log out:

.KJOB

```
JOB 49 USER MCCARRON, M [27,5055]
LOGGED-OFF TTY31 AT 10:37:10 ON 5-DEC-79
RUNTIME: 0:00:00, KCS:5, CONNECT TIME: 0:09:51
DISK READS:63, WRITES:20, BLOCKS SAVED: 4335
```

Then log in once more.

.LOG 27,5055

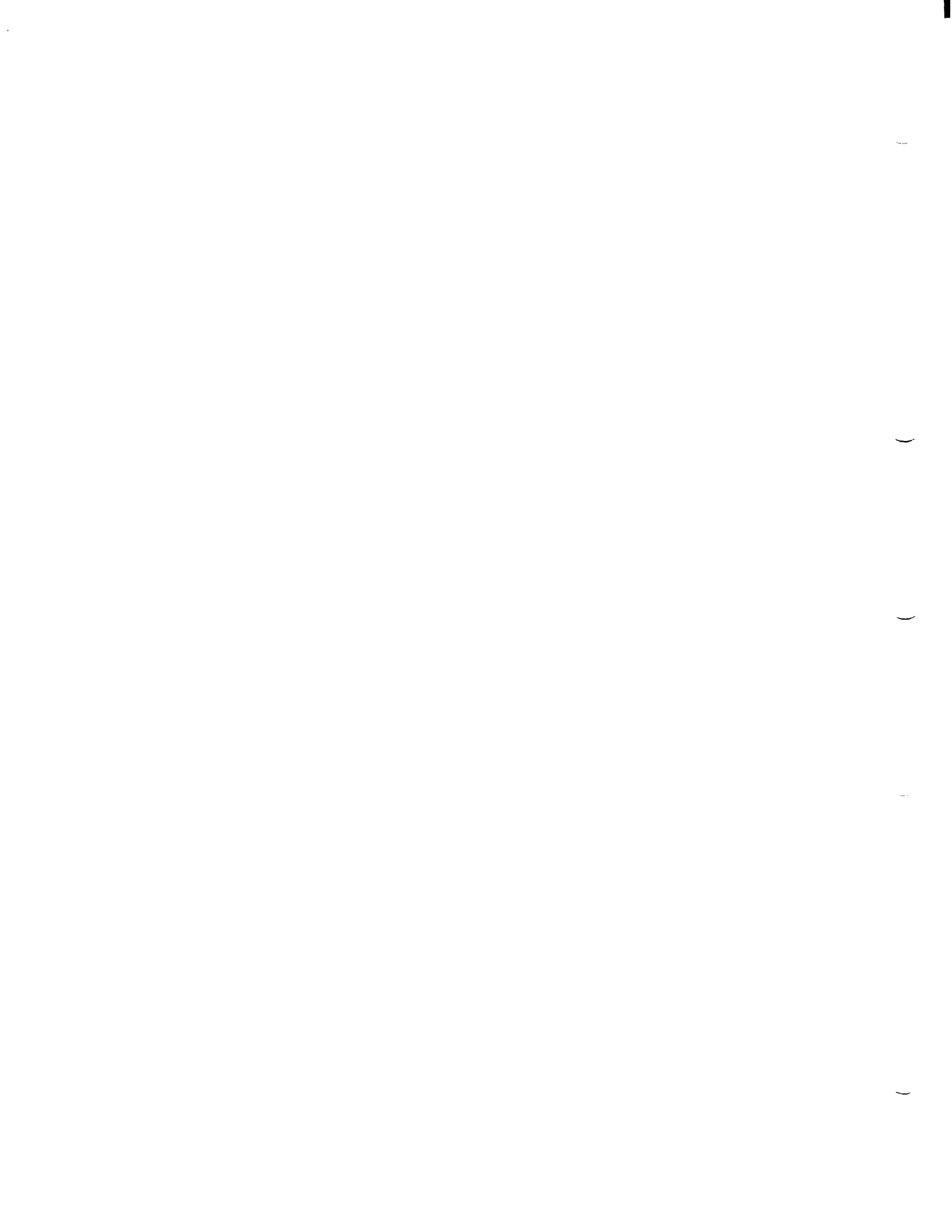
```
JOB 49 RZ024A KL #1026/1042 TTY31
PASSWORD:<RET>
10:40 05-DEC-79 WED
```

Type the command INI TTY to see your terminal characteristics.

.INI TTY

```
RZ024A KL#1026/1042 10:27:27 TTY31 SYSTEM 1026/1042
CONNECTED TO NODE KUL26 (26) LINE #31
JOB 49 USER MCCARRON, M [27,5055]
DSKC: KL1026 SYSTEM DISK DSKC
DSKB: KL1026 SYSTEM DISK DSKB
```

TYPE:VT52	LC	TABS	NOFORM
ECHO	CRLF	WIDTH:80	GAG
DISPLA	FILL:0	NOTAPE	PAGE:24
BLANKS	NOLTMOD	RCVSPEED:2400	XMTSPEED:2400
NODEBREA	NOTIDY	NORTCOMP	NOSLAVE
NOHDX	REMOTE		



APPENDIX C  
COMPILE-CLASS COMMANDS

The six basic COMPILE-class commands are:

```
COMPILE
LOAD
EXECUTE
DEBUG
MAKE
TECO
```

Each COMPILE-class command is described in detail in Chapter 2.

These commands cause the monitor to run the COMPIL program, which decipheres the command and constructs new command strings for the system program (such as, TECO, PIP, FORTRAN). Each time you type the MAKE or TECO commands, the command with its arguments is written as a temporary file in memory or on the disk. Therefore, the system can recall the last file specification you typed. This is an exception to the requirement that the file name must always be specified. For example if you type:

```
.MAKE PROGX.MAC
```

You can then later type the following:

```
.TECO
```

instead of the command line:

```
.TECO PROGX.MAC
```

assuming that you have not issued a TECO command changing the file name in the interim.

The system also writes the COMPILE, LOAD, EXECUTE, and DEBUG commands, with their arguments, in two places:

1. In memory in temporary files written on device TMP:. These are deleted when you log out or when the system crashes.
2. On the disk in a temporary file.

When you use a file specification as the argument to one of these commands, you can use another of these commands with no argument. The stored arguments will be used.

## COMPILE-CLASS COMMANDS

### C.1 INDIRECT COMMANDS (@ CONSTRUCTION)

When you must type many program names and switches, you can put them into a file that eliminates the need for you to retype the names and switches for each compilation. These files are called command files. You can use the @ file construction, which you can type with any COMPILE-class command.

You can specify an @ file at any point in a command line after the first word in the command. In this construction, when you specify a file, you do so by typing its file name, followed by an optional file name extension and project-programmer number. If you omit the extension, the program searches for a command file with a .CMD file name extension. If that file is not found, the program then searches for a command file with a null extension. Then, when the program finds the specified file, it places the information stored in the file in the command string, replacing @file name. If the file is not found, the program prints an error message.

#### Example

If you have a file called FLIST.CMD that contains the following command string:

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

You could replace this command line:

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

with the following command line:

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

You can have command files that contain the @ file construction to a depth of 15 levels. If this process of indirection results in files pointing in a loop, the maximum depth is exceeded, and the program prints an error message:

```
?NESTING TOO DEEP
```

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

1. Spaces are used to delimit words, but are otherwise ignored. Similarly, TABs, vertical TABs, and form feed characters are treated as spaces.
2. To allow long command strings, command terminators (such as RETURN, ESCAPE) are ignored if the first nonblank character after a sequence of command terminators is a comma. Otherwise, the command terminators are treated as commas by the COMPILE-class commands.
3. Blank lines are completely ignored.
4. Comments can be included in command files by preceding the comment with a semicolon; text from the semicolon to the end of the line is ignored.
5. If command files are sequenced, the sequence numbers are ignored.



## COMPILE-CLASS COMMANDS

### C.2 THE + CONSTRUCTION

The + construction specifies that the files to the right of the plus sign are combined with the first file you specified. Therefore, all files you specify become one file in the compilation and not separate files.

This construction is useful when the first file is a subroutine of the other files you specified.

When you use the + construction, the compiler produces a single relocatable binary file from a collection of input source files. To construct a single program from several input files, you can name one input file FIRST.MAC, another MIDDLE.MAC, and a third LAST.MAC. You can then specify the following command line:

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

After this, the compiler produces one binary file named LAST from the three source files, FIRST, MIDDLE, and LAST. This construction allows you to use one input file as part of several different compilations. For example, you could later use the FIRST.MAC file with SECOND.MAC and THIRD.MAC to obtain a different binary file. The + construction permits you to maintain material in a single file that is common to more than one compilation.

The compiler gives the name of the last input file in the string to any output file (for example, .REL, .CRF, .LST). Therefore, in the previous examples, the output files would be called LAST and THIRD, respectively. Note that you can include device names, extensions, and project-programmer numbers in any + construction. Therefore, the following is a valid command string:

```
.COMPILE FIRST.MAC[27,4072]+SECOND.MAC+THIRD.MAC[35,234]
```

### C.3 THE = CONSTRUCTION

Usually the compiler makes the name of the relocatable binary file the same as that of the source file, with the extension specifying the difference. You can override this action by using the = construction. The = construction allows you to specify the names of the output files. For example, if you need a binary file called BINARY.REL from a source program called SOURCE.MAC, you can type the following command line:

```
.COMPILE BINARY=SOURCE
```

You can also use this construction to specify an output name for a file produced by using the + construction. To give the name WHOLE.REL to the binary file the compiler assembles with PART1.MAC and PART2.MAC, you can type the following command line:

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

To change the file name of the output file is the most common use of the = construction. However, you can also use it to change any other default condition. The default condition for compiler output is DSK:SOURCE.REL[dir]. If you would like output on DTA3 with the file name FILEX instead of the defaults, you could type the following command line:

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

## COMPILE-CLASS COMMANDS

### C.4 THE < > CONSTRUCTION

The < > (angle bracket) construction can be used in COMPILE, LOAD, EXECUTE, and DEBUG commands only. You use it to cause programs listed within the angle brackets to be assembled with the same parameter file. If you also use the + construction, it must appear before the < > construction. To assemble LPTSER.MAC, PTPSER.MAC, and PTRSER.MAC with the PAR.MAC parameter file, you could type the following command line:

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

However, using the < > construction you could type the following line:

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

The following command string construction is invalid:

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

### C.5 DEFAULT COMPILER

When you name a file with a standard file extension (.MAC,.CBL,.FOR,.ALG), the compiler implied by the extension will be used to compile the program. Standard file extensions are listed in Appendix D. You cannot over-ride the compiler that is implied by the file extension. The COBOL compiler will compile a file called DATPRO.CBL. Files without a recognizable compiler extension are compiled by the default compiler, normally FORTRAN.

If you have a file with a non-standard file extension, you can control the setting of the default compiler (FORTRAN) by including switches in the COMPILE, LOAD, EXECUTE, or DEBUG command string. The descriptions of the COMPILE, LOAD, EXECUTE, and DEBUG commands in Chapter 2 include the switches you can use to change the default compiler.

In the following examples, the installation has chosen FORTRAN as the default compiler. If you issue the following command:

```
.COMPILE NOEXT
```

the file named NOEXT will be compiled by FORTRAN. The following command:

```
.COMPILE FILEZ.MIN
```

causes the file named FILEZ.MIN to be compiled by FORTRAN because MIN is not a recognizable compiler extension. The following command:

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

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

#### NOTE

By setting the appropriate assembly switches you can indicate SNOBOL or MACYll as the compiler. However, these switches and compilers are not supported.

## COMPILE-CLASS COMMANDS

The switches that change the default compiler can be temporary or permanent. (Refer to Section 1.7.4.)

### C.6 COMPILER SWITCHES

It is occasionally necessary for you to pass switches to the assembler or compiler 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:

1. Binary output file specification.
2. Listing file specification.
3. Source file specification.

To include switches with these files, you must do the following:

1. Group the switches according to each related source file, if you use the + construction.
2. Group the switches according to the three types of files for each source file; the order of the groups of switches is
  - a. Binary
  - b. Listing
  - c. Source
3. Separate the groups of switches by commas for each source file.
4. Enclose all switches for each source file within one set of parentheses.
5. Place each parenthetical string immediately after the source file to which it refers.

The COMPIL program interprets the groups of switches according to these rules:

1. The switches immediately before a closing right parenthesis are source (SSSS).
2. The switches before the first comma are binary (BBBB,,).
3. The switches before the second comma are listing (,LLLL,).
4. The order of the switches is BBBB, LLLL, SSSS.
5. The individual switches are separated by spaces.

#### Examples

(SSSS)	Source switch
(BBBB,,)	Binary switch
(BBBB,LLLL,SSSS)	Binary, list, and source switches
(,,SSSS)	Source switch(es)
(,LLLL,)	Listing switch(es)
(BBBB,,SSSS)	Binary and source switches
(BBBB,LLLL,)	Binary and list switches
(,LLLL,SSSS)	List and source switches

## COMPILE-CLASS COMMANDS

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

Table C-1  
Processor Switches

Processor	Binary	Listing	Source	Meaning		
ALGOL	D			Set dynamic storage region for your own array (known as the heap).		
			E	The source file has line numbers in columns 73 to 80.		
			L	List the source program.		
			N	Suppress output of error messages on the terminal.		
			Q	Delimit the words in quotes.		
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	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	M	Print a map showing the parameters of the user-defined items.
			N		Suppress output of source errors on the terminal.	
P			Do not generate trace calls and symbols.			

COMPILE-CLASS COMMANDS

Table C-1 (Cont.)  
Processor Switches

Processor	Binary	Listing	Source	Meaning
COBOL (Cont.)	R			Produce a two-segment object program. The high segment contains the resident sections of the Procedure division; the low segment contains everything else. When the object program is loaded, LIBOL is added to the high segment.
	S	S	S	The source file has sequence numbers in columns 1 through 6 and comments starting at column 73.
	W	W	W	Rewind the magnetic tape.
	Z	Z	Z	Zero the DECTape directory.
FORTRAN-10			C	Generate a file that is to 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.
	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 on the terminal.
	NOW	NOW	Suppress output of warning messages on the terminal.	
	O		S	Perform optimization of global symbols when producing processor code.
			S	Perform compilation, checking for syntax errors only.
MACRO	A	A	A	Advance the magnetic tape reel by one file.
	B	B	B	Backspace the magnetic tape reel by one file.

COMPILE-CLASS COMMANDS

Table C-1 (Cont.)  
Processor Switches

Processor	Binary	Listing	Source	Meaning	
MACRO (Cont.)		C		Produce a listing file in a format acceptable as input to CREF.	
		E		List the 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 test in macro and repeat the expansion (SALL).	
		N		Suppress output of error messages on the terminal.	
		O	O	O	Allow literals to occupy only one file.
		P			Increase the size of the pushdown list.
		Q	Q	Q	Suppress questionable (Q) error indications on the listing.
			S		Suppress the listing.
		T	T	T	Skip to the logical end of the magnetic tape.
			X		Suppress all macro expansions.
		Z	Z	Zero the DECTape directory.	

## COMPILE-CLASS COMMANDS

The following are examples of the use of switches:

### Example

<code>.DEBUG TEST(,N,)</code>	Suppress error message during assembly.
<code>.COMPILE OUTPUT=MTA0:(S,M,W)/L</code>	Rewind the magtape (W), compile the first file, produce binary output for the K110 (S), and include the MACRO coding in the output listing file (M). Output files are given the names OUTPUT.REL and OUTPUT.LST.
<code>.COMPILE/MACRO A=MTA0:(,Q,W)/L</code>	Rewind the magtape (W), compile the first file, and suppress Q (questionable) error indications in the listing. When a binary switch is not present, the delimiting comma must appear.
<code>.COMPILE/MACRO A=MTA0:(,Q,)/L</code>	Compile the file at the current position of the tape and suppress Q error indications on the listing. When the source and binary switches are not present, the delimiting commas must appear.
<code>.COMPILE FOO (NOWARN DEBUG)</code>	Compile the file named FOO using the F10 source switches NOWARN and DEBUG.

### C.7 LINK SWITCHES

One linking loader is available to you on TOPS-10: the LINK program. (See the LINK Programmer's Reference Manual for complete information.) In complex loading procedures, it might be necessary to pass switches to LINK to direct its operation. The percent (%) character accomplishes this passing of switches.

The LINK switch specification, preceded and followed by a delimiter, follows the % character. The delimiter can be any character; however, you must be careful that the character you use does not have a specific meaning to COMPIL. The @ character indicates an indirect command file, and the semicolon (;) causes the remainder of the line to be treated as a comment and, therefore, is ignored. The recommended delimiter is a single or double quotation mark. The beginning and ending delimiter must be the same character.

A LINK specification consists of the switch name and, optionally, a keyword and a value. See the LINK Reference Manual for switch formats.





APPENDIX D  
STANDARD SYSTEM NAMES

D.1 FILE NAME EXTENSIONS

Table D-1 lists the file name extensions that have specific meanings to TOPS-10.

Table D-1  
File Name Extensions

File Name 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.
Bl0	Source	Source file in BLIS10.
Bl1	Source	Source file in BLISS-11.
BAC	Object	Reserved for output from the BASIC compiler.
BAK	Source	Backup file from TECO or SOS.
BAS	Source	Source file in BASIC language.
BCM	ASCII	Listing file created by FILCOM (binary compare).
BCL	Source	Source file in BCPL language.
BIN	Binary	Binary file.
BLB	ASCII	Blurb file.
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 nonresident 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.

STANDARD SYSTEM NAMES

Table D-1 (Cont.)  
File Name Extensions

File Name Extension	Type of File	Meaning
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	Batch control file.
DAE	Binary	Default output for DAEMON-taken core dumps.
DAT	ASCII, Binary	Data (FORTRAN) file.
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 file name.
ERR	ASCII	Error message file.
EXE	Object	Executable program.
F4	Source	Source file in Fortran-40 (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.
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.

STANDARD SYSTEM NAMES

Table D-1 (Cont.)  
File Name Extensions

File Name Extension	Type of File	Meaning
MEM	ASCII	Memorandum file typically output from RUNOFF.
MIC	ASCII	MIC command file.
MID	Source	Source file in MIDAS (MIT Assembler) language.
MIM	Binary	Snapshot of MIMIC simulator.
MSB	Object	Music compiler binary output.
MUS	Source	Music compiler input.
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.
Pl1	Source	Source program in MACY11 language.
PAK	ASCII	Files compressed by PACK.TEC to save disk space.
PAL	Source	Source file in PAL10 (PDP-8 assembler).
PL1	Source	Source file in PL1 language.
PLM	ASCII	Program Logic Manual.
PLO	Binary	Compressed plot output.
PLT	ASCII	Spooled output for plotter.
PPL	Source	Source file in PPL language.
PTP	ASCII, Binary	Spooled 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.
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).
RNB	ASCII	RUNOFF input for producing a .BLB file.
RNC	ASCII	RUNOFF input for producing a .CCO file.
RND	ASCII	RUNOFF input for producing a .DOC file.
RNE	ASCII	RUNOFF input for error message text.
RNH	ASCII	RUNOFF input for producing a .HLP file.
RNL	ASCII	RUNOFF input for program logic manual.
RNM	ASCII	RUNOFF input for producing a .MAN file.
RNO	ASCII	Programming specifications in RUNOFF input to produce .MEM file.
RNP	ASCII	RUNOFF input for producing a .OPR file.
RNS	ASCII	RUNOFF input for text file of standards.
RSP	ASCII	Script response time log file.
RSX	All	Files for RSX-11D.

STANDARD SYSTEM NAMES

Table D-1 (Cont.)  
File Name Extensions

File Name Extension	Type of File	Meaning
RTB	Object	Read-in mode (RIM10B) format file (PIP).
SAI	Source	Source file in SAIL language.
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).
SMP	Source	Source file in SIMPLE language.
SNO	Source	Source file in SNOBOL language.
SNP	ASCII	Snapshot of disk by DSKLST.
SOS	ASCII	SOS command file.
SPC	ASCII	Corrected file for SPELL program.
SPD	ASCII	Dictionary for SPELL program.
SPM	ASCII	File of misspelled words for SPELL program.
SPU	ASCII	File of upper case words for SPELL program.
SPX	ASCII	File of exception (error) lines for SPELL program.
SRC	ASCII	Source files.
STD	ASCII	Standards.
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.
TPB	ASCII	Typeset input for producing a .BLB file.
TPC	ASCII	Typeset input for producing a .CCO file.
TPD	ASCII	Typeset input for producing a .DOC file.
TPE	ASCII	Typeset input for producing error message text.
TPH	ASCII	Typeset input for producing a .HLP file.
TPL	ASCII	Typeset input for producing a logic manual.
TPM	ASCII	Typeset for producing a .MAN file.
TPO	ASCII	Typeset input for producing a programming specification.
TPP	ASCII	Typeset input for producing an .OPR file.
TST	All	Test data.
TXT	ASCII	Text file.

STANDARD SYSTEM NAMES

Table D-1 (Cont.)  
File Name Extensions

File Name Extension	Type of File	Meaning
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.
XPN	Object	Expanded save file (FILEX and LINK-10).
Zxx	ASCII	Edit original file (all xx).

## STANDARD SYSTEM NAMES

### D.2 RESERVED PROJECT-PROGRAMMER NUMBERS

Table D-2 is a list of the project-programmer numbers that are allocated for specific functions in TOPS-10.

Table D-2  
Project-Programmer Numbers

Number	Use of Number
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	User maintained library (device 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.
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:).
5,20	NELIAC (device NEL:).
5,21	DUMP (device DMP:).
5,22	POP2 (device POP:).
5,23	Test library (device TST:).
6,*	Field service and hardware diagnostics.
7,7	Software acceptance.
10,1	Special system programming storage region containing copies of SYS:CRASH.EXE (device XPN:).
10,6	Software distribution.
10,7	Software distribution (device DEC:).

STANDARD SYSTEM NAMES

D.3 ERSATZ DEVICE NAMES

Table D-3 lists ersatz device names that are predefined in TOPS-10.

To avoid confusion, it is recommended that these names not be used for private file structures.

Table D-3  
Ersatz 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
DEC:	DEC supplied software	[10,7]	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
NUL:		None	
OLD:	Old system library	[1,3]	System
PUB:	User maintained library	[1,6]	System
REL:	Rel file library	[5,11]	System
RNO:	RUNOFF library	[5,12]	System
SNO:	SNOBOL library	[5,13]	System
STD:	Standard Software	[1,4]	System
SYS:	System library	[1,4]	System
TED:	Text editor library	[5,10]	System
TST:	Test library	[5,23]	System
UNV:	MACRO universal library	[5,17]	System
XPN:	Crash library	[10,1]	System





APPENDIX E

CARD CODES

Table E-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.

CARD CODES

Table E-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
0	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	1-8-6	{	173	12-0
<	74	12-8-4	}	174	12-11
=	75	8-6	~	175	11-0
>	76	0-8-6	DEL	176	11-0-1
?	77	0-8-7		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.

## CARD CODES

Table E-2  
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 and 140-177 are the same as in ASCII.



APPENDIX F  
TEMPORARY FILES

The temporary files in Table F-1 are used by various programs in the DECsystem-10 computing system. These files are in the following form:

nnnxxx.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 F-1  
Temporary Files

Name	Meaning
nnnALG.TMP	Read by ALGOL and contains one line for each program to be compiled. It may also contain the command NAME! that causes ALGOL to transfer control to the named program.
nnnAS1.TMP nnnAS2.TMP nnnAS3.TMP	Written, read, and deleted by COBOL and contains input to the COBOL assembler.
nnnBL1.TMP	Read by BLISS and contains one line for each program to be compiled.
nnnCOB.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.
nnnCPY.TMP	Written, read, and deleted by COBOL and contains copies of source files with library routines inserted.
nnnCRE.TMP	Read by CREF and contains commands for each file that 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.
nnnDAE.TMP	Written by DAEMON to be read by DUMP.
nnnDMP.TMP	Read by DUMP as an input command file.

## TEMPORARY FILES

Table F-1 (Cont.)  
Temporary Files

Name	Meaning
nnnEDS.TMP	Used by COMPIL to store the arguments of the most recent EDIT, CREATE, TECO, or MAKE command.
nnnEDT.TMP	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  S file.ext [p,p] <span style="border: 1px solid black; border-radius: 50%; padding: 0 2px;">ESC</span>  For TECO or EDIT commands, it contains the command  S file.ext [p,p]
nnnERA.TMP	Written, read, and deleted by COBOL and is the error file.
nnnFAL.TMP	Read by FAIL and contains one line for each program to be compiled.
nnnFOR.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.
nnnGEN.TMP	Written, read, and deleted by COBOL and contains the output of syntax processing.
nnnKJO.TMP	Read by KJOB as an input command file.
nnnLGO.TMP	Read by LOGOUT as an input command file.
nnnLHC.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.
nnnLIN.TMP	Created by LINED and contains output file until the rename process.
nnnLIT.TMP	Written, read, and deleted by COBOL and contains copy of the literal pool.
nnnLLC.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.
nnnLLS.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.
nnnLNK.TMP	Read by LINK-10 and contains commands necessary for loading.
nnnLOA.TMP	Read by LOADER and contains commands necessary for loading.

TEMPORARY FILES

Table F-1 (Cont.)  
Temporary Files

Name	Meaning
nnnMAC.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.
nnnP11.TMP	Read by MACY11 (the PDP-11 assembler for the PDP-10) and contains one line for each program to be assembled.
nnnPLS.TMP	Read by PLEASE as an input command file.
nnnPIP.TMP	Read by PIP and contains commands to implement the COMPIL-class commands that run PIP.
nnnQUE.TMP	Read by QUEUE as an input command file.
nnnRNO.TMP	Read by RUNOFF and contains commands for each file which has produced a RUNOFF listing on the disk.
nnnS01.TMP	Written, read and deleted by COBOL and contains the intermediate sort results of the data.
nnnSVC.TMP	Used by COMPIL to store the arguments of the most recent COMPIL, LOAD, EXECUTE, or DEBUG command.
nnnSNO.TMP	Read by SNOBOL and contains one line for each program to be compiled.
nnnTEC.TMP	Created by TECO and contains output file until the rename process.
nnnTMP.TMP	Created by LINED during the rename process.
nnnXFO.TMP	Created by FILEX as a result of the Q switch on the output side.
nnnXFR.TMP	Created by FILEX as a result of the Q switch on the input side.





APPENDIX G

SIXBIT/ASCII CHARACTER CODES

SIXBIT	Character	ASCII 7-Bit	SIXBIT	Character	ASCII 7-Bit	Character	ASCII 7-Bit
00	Space	040	40		100	,	140
01	!	041	41	A	101	a	141
02	"	042	42	B	102	b	142
03	#	043	43	C	103	c	143
04	\$	044	44	D	104	d	144
05	%	045	45	E	105	e	145
06	&	046	46	F	106	f	146
07	'	047	47	G	107	g	147
10	(	050	50	H	110	h	150
11	)	051	51	I	111	i	151
12	*	052	52	J	112	j	152
13	+	053	53	K	113	k	153
14	,	054	54	L	114	l	154
15	-	055	55	M	115	m	155
16	.	056	56	N	116	n	156
17	/	057	57	O	117	o	157
20	0	060	60	P	120	p	160
21	1	061	61	Q	121	q	161
22	2	062	62	R	122	r	162
23	3	063	63	S	123	s	163
24	4	064	64	T	124	t	164
25	5	065	65	U	125	u	165
26	6	066	66	V	126	v	166
27	7	067	67	W	127	w	167
30	8	070	70	X	130	x	170
31	9	071	71	Y	131	y	171
32	:	072	72	Z	132	z	172
33	;	073	73	[	133	{	173
34	<	074	74	\	134	\	174
35	=	075	75	]	135	}	175
36	>	076	76	↑	136	~	176
37	?	077	77	←	137	Delete	177



## INDEX

+ constructions, C-3  
 .CMD files, C-2

.EXE files, 2-176

.REL files, 2-115

< > constructions, C-4  
 = constructions, C-3  
 @ constructions, C-2

Access dates, 2-71  
 ACCESS.USER file, 1-22  
 Accessing,  
   DECTapes, 2-108  
   files, 1-12, 1-20, 1-22,  
     2-190  
   tapes, 2-11  
   the system, 2-146  
 Active search list, 2-265  
 Argument,  
   core, 2-33  
   node-id, 2-174  
 Arguments,  
   command, 1-8  
   date-time, 1-9  
 Arranging files, 1-25  
 ASCII card codes, E-1  
 Assembler switches, C-5  
 Assembling programs, C-4  
 ASSIGN command, 1-16, 2-5  
 Assigning,  
   core, 2-33  
 ATTACH command, 2-7  
 Authorizing users, 1-1

Backing up files, 2-11  
 BACKSPACE command, 2-10  
 Backup,  
   files, 2-70  
 BACKUP program, 2-11  
 Batch,  
   command interpreter, 1-2  
   jobs, 1-1, 1-2  
   mode, 1-2

Batch jobs,  
   starting, 2-273  
 Binary files, 2-22, 2-115  
 Bits,  
   depositing, 2-45  
   schedule, 2-221  
 Blocksize,  
   setting tape, 2-225  
 Break points,  
   setting, 2-226  
 Buffer number,  
   changing, 2-232

Calling files, 1-12  
 Card codes,  
   ASCII, E-1  
   DEC-026, E-4  
 Card punch output, 2-35  
 Card reader,  
   setting, 2-229  
 CCONTINUE command, 2-18  
 Changing,  
   buffer number, 2-232  
   directory paths, 2-266  
   nodes, 2-144, 2-239  
   processors, 2-230  
   protection codes, 2-190  
   search lists, 2-265  
   terminals, 2-211  
 Characters,  
   deleting, 1-4  
   wildcard, 1-22  
 Checksum, 2-71  
 Clearing directories, 2-294  
 CLOSE command, 2-21  
 Closing files, 2-113  
 COBOL programs,  
   debugging, 2-59  
 Codes,  
   ASCII card, E-1  
   DEC-026 card, E-4  
   SIXBIT/ASCII, G-1  
 Combining,  
   files, 2-31, C-3  
   programs, 2-158  
 Command,  
   arguments, 1-8  
   format, 1-7  
   functions, A-1  
   options, 1-10  
   strings, 1-7  
   switches, 1-10  
   syntax, 1-7

INDEX (CONT.)

- Command files,
  - indirect, C-2
- Command language,
  - interpreter, 1-2
- Commands,
  - COMPILE-class, C-1
  - ending, 1-8
  - terminating, 1-2, 1-8
- Comments, 1-11
- Communicating,
  - with the operator, 2-183
  - with the system, 1-1
- Comparing,
  - files, 2-99
  - symbols, 2-119
- COMPILE command, 2-22
- COMPILE-class commands, C-1
- Compiler switches, C-5
- Compilers,
  - default, C-4
  - standard, 2-22
- Configuration,
  - system, 2-174
- Constructions,
  - +, C-3
  - < >, C-4
  - =, C-3
  - @, C-2
  - wildcard, 1-22
- CONTINUE program, 2-29
- Continuing,
  - jobs, 2-133
  - lines, 1-8
  - programs, 2-18
  - terminal output, 1-6
- Control-C, 1-4
- Control-O, 1-6
- Control-Q, 1-6
- Control-R, 1-5
- Control-S, 1-6
- Control-T, 1-6
- Control-U, 1-5
- Control-W, 1-4
- COPY command, 2-31
- Core,
  - assigning, 2-33
  - examining, 2-89
- CORE command, 2-33
- Core image files,
  - creating, 2-176, 2-219
  - writing, 2-47
- Correcting typing errors,
  - 1-4, 1-5
- CPU time limits, 2-250
- CPUNCH command, 2-35
- Creating,
  - core image files, 2-219
  - directories, 2-38
  - executable files, 2-176
- Creating (Cont.)
  - files, 2-157
  - queue entries, 2-35, 2-184, 2-188, 2-193, 2-273
- CREDIR program, 2-38
- CREF command, 2-40
- Cross-referenced listings,
  - 2-40, 2-60, 2-94, 2-139
- CSTART command, 2-42
- CTRL/C, 1-4, 2-123
- CTRL/O, 1-6
- CTRL/Q, 1-6
- CTRL/R, 1-5
- CTRL/S, 1-6
- CTRL/T, 1-6, 2-288
- CTRL/U, 1-5
- CTRL/W, 1-4
- Current virtual page limit,
  - 2-260
- D command, 2-45
- DAEMON program, 2-47
- DAEMON-written files, 2-47
- Data,
  - depositing, 2-45
  - storing, 1-12
- Date-time arguments, 1-9
- DAYTIME command, 2-46
- DCORE command, 2-47
- DDT command, 2-53
- DEASSIGN command, 2-57
- Deassigning devices, 2-113
- DEBUG command, 2-59
- Debugging programs, 2-53, 2-226
- DEC-026 card codes, E-4
- DECTape,
  - delimiters, 2-136
  - files, 1-12
  - identifiers, 2-136
- DECTapes,
  - accessing, 2-108
  - labelling, 2-136
- Default,
  - compilers, C-4
  - directories, 1-25
- Defaults,
  - system, 1-22
- DELETE,
  - command, 2-67
  - key, 1-4
- Deleting,
  - characters, 1-4
  - files, 2-294
  - lines, 1-5
  - words, 1-4

INDEX (CONT.)

- Density,
  - setting tape, 2-235
- Depositing information,
  - 2-45
- DETACH command, 2-69
- Detaching jobs, 2-7
- Device,
  - errors, 2-133
  - names, 1-13
- Device names,
  - ersatz, 1-18
  - generic, 1-14
  - logical, 1-16
  - physical, 1-15
  - user-defined, 1-16
- Devices,
  - assigning, 2-5
  - assigning network, 2-5
  - available, 2-215
  - deassigning, 2-57, 2-113
  - disk, 1-16
  - ersatz, 2-38
  - locating, 2-292
  - network, 1-15
  - output, 1-13
  - peripheral, 1-13
  - reassigning, 2-209
  - restricted, 1-17, 2-5
  - spooling, 2-150, 2-247
- Diagnostic messages, 3-1
- Directories,
  - clearing, 2-294
  - creating, 2-38
  - default, 1-25
  - expunging, 2-294
  - master-file, 1-19
  - specifying, 1-23
  - sub-file, 1-19, 1-25, 2-38
  - user-file, 1-19, 1-25, 2-38
- Directory,
  - areas, 1-1, 1-19, 1-25
  - names, 1-19
  - paths, 1-25
- DIRECTORY command, 2-70
- Directory paths,
  - changing, 2-266
- DISABLE command, 2-79
- Disk,
  - devices, 1-16
  - I/O, 2-237
  - quota, 2-206, 2-236
  - usage statistics, 2-86, 2-150
- Disk priority,
  - setting, 2-237
- Displaying files, 2-286
- DSK command, 2-86
- DUMP command, 2-88
- E command, 2-89
- Editing files, 2-281
- ENABLE command, 2-91
- End-of-file marks, 2-92
- Ending,
  - commands, 1-8
  - jobs, 1-1, 2-134
- Entering user level, 2-30, 2-124
- Entries creating,
  - queue, 2-273
- EOF command, 2-92
- Error messages, 3-1
- Error reporting,
  - tape, 2-245
- Errors,
  - correcting typing, 1-4, 1-5
  - device, 2-133
- Ersatz,
  - device names, 1-18
- Ersatz devices, 2-38, D-7
- Examining,
  - core, 2-89
  - memory, 2-89
- Executable files,
  - creating, 2-176
- EXECUTE command, 2-93
- Execution,
  - starting, 2-217, 2-271
  - stopping, 2-123
- Expunging directories,
  - 2-294
- Extensions,
  - standard file, C-4
- Fatal errors, 3-2
- FENCE, 2-265
- FILCOM program, 2-99
- File,
  - extensions, 1-19
  - names, 1-19
  - structures, 1-16, 1-24
- FILE command, 2-108
- File Daemon, 1-22
- File extensions,
  - standard, C-4, D-1
- File names,
  - recalling, C-1
- Files, 1-12
  - .CMD, C-2
  - .EXE, 2-176
  - .REL, 2-115

INDEX (CONT.)

- Files (Cont.)
  - accessing, 1-12, 1-20, 1-22
  - arranging, 1-25
  - backup, 2-70
  - binary, 2-22, 2-115
  - calling, 1-12
  - closing, 2-113
  - combining, 2-31, C-3
  - comparing, 2-99
  - copying, 2-31
  - creating, 2-157
  - DAEMON-written, 2-47
  - DECTape, 1-12
  - deleting, 2-67, 2-294
  - displaying, 2-286
  - DUMP, 2-88
  - editing, 2-281
  - indirect command, C-2
  - library, 2-115
  - listing, 2-137
  - naming, 1-19
  - option, B-1
  - printing, 2-137
  - protecting, 1-20, 2-190
  - renaming, 2-213
  - restoring, 2-11
  - specifying, 1-12, 1-22, 1-23, C-3
  - SWITCH.INI, B-1
  - temporary, F-1
  - transferring, 2-31
- Filler characters, 2-255
- Find Files, 2-73
- FINISH command, 2-113
- Floating point simulation, 2-238
- Format,
  - command, 1-7
  - switch, 1-10
- FORTTRAN programs,
  - optimizing, 2-62
- FUDGE command, 2-115
- Functions,
  - command, A-1
  
- Generic device names, 1-14
- GET command, 2-117
- GETTAB tables, 2-49
- GLOB program, 2-119
- Global switches, 1-11
- Global symbols,
  - listing, 2-119
- Guidelines,
  - paging, 2-243
  
- HALT command, 2-123
- Halting programs, 2-123
- Hard errors, 2-245
- HELP command, 2-125
- High-priority run queue, 2-241
- Host nodes, 2-174
  
- I/O,
  - disk, 2-237
  - terminating, 2-21, 2-113
- Identifiers,
  - DECTape, 2-136
- Indirect command files, C-2
- Information,
  - depositing, 2-45
  - job, 1-6, 2-49, 2-182, 2-262, 2-288
  - network, 2-169, 2-174
  - obtaining, 2-125
  - system, 2-275
- INITIA command, 2-128
- Initiating jobs, 2-146
- Input queue, 2-273
- Input queues, 2-193
- Interactive mode, 1-2
- Interpreter,
  - batch command, 1-2
  - command language, 1-2
- Interrupting programs, 1-4, 2-123
  
- JCONTINUE command, 2-133
- Job,
  - information, 2-49, 2-182, 2-288
  - number, 2-182
  - search list, 1-24, 2-265
  - statistics, 2-262
  - status, 1-6
- Job data area, 2-212
- Job status, 1-6
- Jobs, 1-1
  - attaching, 2-7
  - batch, 1-1, 1-2
  - continuing, 2-133
  - detaching, 2-7, 2-69
  - ending, 1-1, 2-134
  - initiating, 2-146
  - killing, 2-134
  - starting, 1-1, 2-146
  - temporary, 2-7
  - terminating, 2-134

INDEX (CONT.)

- Jobs (Cont.)
  - transferring, 2-211
- Key,
  - DELETE, 1-4
  - RUBOUT, 1-4
- KJOB command, 2-134
- LABEL command, 2-136
- Level,
  - monitor, 1-2, 1-4
  - program, 1-3
  - user, 1-3
- Libraries,
  - system, 1-18, D-7
- Library files, 2-115
- Limit,
  - physical page, 2-150
- Line continuation, 1-8
- Line printer, 2-188
- Line printer output, 2-137
- Line-editing, 1-4
- LINK program, C-9
- LIST command, 2-137
- Listing,
  - files, 2-137
  - global symbols, 2-119
- Listings,
  - cross-referenced, 2-60, 2-94, 2-139
- LOAD command, 2-138
- Loading programs, 2-117, 2-138
- Local switches, 1-11
- LOCATE command, 2-144
- Locating devices, 2-292
- Logging,
  - in, 2-146
  - out, 2-134
- Logical device names, 1-16
- Logical names,
  - assigning, 2-5
- LOGIN command, 2-146
- Magnetic tapes,
  - see Tapes, G-2
- MAKE command, 2-157
- Marks,
  - end-of-file, 2-92
- Master-file directories,
  - 1-19
- MCR, 2-174
- Memory,
  - buffers, 2-232
  - examining, 2-89
  - pages, 2-150, 2-152, 2-243, 2-260
- Memory space,
  - assigning, 2-33
- MERGE command, 2-158
- Messages,
  - sending, 2-223
  - warning, 3-2
- MFD, 1-19
- MIC commands, 2-161
- Mode,
  - batch, 1-2
  - interactive, 1-2
- Monitor, 1-1
- Monitor level, 1-2, 1-4
- Moving tapes, 2-10, 2-270
- Naming files, 1-19
- Network,
  - devices, 1-15
  - information, 2-169, 2-174
  - topology, 2-171
- NETWORK command, 2-169
- Network devices,
  - assigning, 2-5
- NODE command, 2-174
- Node-id argument, 2-174
- Nodes,
  - changing, 2-144, 2-239
  - host, 2-174
- NSAVE command, 2-176
- NSSAVE command, 2-176
- Obtaining information,
  - 2-125
- Operator,
  - communicating with the, 2-183
- Operator coverage, 2-221
- OPR:, 2-223
- OPSER program, 2-177
- Optimizing FORTRAN programs,
  - 2-62
- Option files, B-1
- Options,
  - command, 1-10
- Output,
  - devices, 1-13
- Output queues, 2-193
- Overflow,
  - disk, 2-236

INDEX (CONT.)

- Page limit,
  - physical, 2-150, 2-243
  - virtual, 2-152
- Pages,
  - memory, 2-150, 2-152
- Paging guidelines, 2-243
- Paper tape punch output,
  - 2-284
- Parameter files, C-4
- Passive search list, 2-265
- Passwords, 1-1
- Paths,
  - directory, 1-25
  - specifying, 2-146
- Peripheral devices, 1-13
- Permanent switches, 1-11
- Physical,
  - device names, 1-15
  - page limit, 2-150, 2-243
- PJOB command, 2-182
- PLEASE command, 2-183
- PLOT command, 2-184
- Plotter output, 2-184
- PPNs, 1-1
  - reserved, D-6
- PRESERVE command, 2-186
- PRINT command, 2-188
- Printing files, 2-137
- Priority,
  - setting disk, 2-237
- Privileged commands, 2-230,
  - 2-237, 2-241
- Privileges,
  - disabling, 2-79
  - enabling, 2-91
- Processor switches, C-6
- Processors,
  - changing, 2-230
- Programs,
  - assembling, C-4
  - combining, 2-158
  - compiling, 2-22
  - continuing, 2-18, 2-29
  - debugging, 2-53, 2-59,
    - 2-226
  - debugging COBOL, 2-59
  - executing, 2-93
  - halting, 2-123
  - interrupting, 1-4, 2-123
  - loading, 2-117, 2-138
  - merging, 2-158
  - optimizing FORTRAN, 2-62
  - reinitializing, 2-212
  - restarting, 2-212
  - running, 1-3, 2-93, 2-217
  - running system, 2-207
  - starting, 2-18, 2-42,
    - 2-271
  - storing, 1-12
- Project-programmer numbers,
  - 1-1
- Prompts, 1-2, 1-3
- PROTECT command, 2-190
- Protecting,
  - files, 1-20, 2-186, 2-190,
    - 2-233
- Protection codes, 1-20
  - changing, 2-190
- Pseudo-terminals, 2-177
- PTYs, 2-177
- QUEUE command, 2-193
- Queue entries,
  - creating, 2-35, 2-184,
    - 2-188, 2-193, 2-273
- Queues,
  - input, 2-193
  - output, 2-193
- QUOLST program, 2-206
- Quota,
  - disk, 2-206, 2-236
- R acommand, 2-207
- REASSIGN command, 2-209
- REATTACH program, 2-211
- Recalling file names, C-1
- REENTER command, 2-212
- RENAME command, 2-213
- Reprinting lines, 1-5
- Reserved PPNs, 1-1, D-6
- RESOURCES command, 2-215
- Restarting programs, 2-212
- Restoring files, 2-11
- Restricted devices, 1-17,
  - 2-5
- REWIND command, 2-216
- RUBOUT key, 1-4
- RUN command, 2-217
- Run queue,
  - high-priority, 2-241
- Run time, 2-250, 2-282
- Running,
  - programs, 1-3, 2-93,
    - 2-217
  - subjobs, 2-177
  - system programs, 2-207
  - TECO, 2-157
- SAVE command, 2-219
- SCHED command, 2-221
- Schedule bits, 2-221



INDEX (CONT.)

Search lists,  
     changing, 2-265  
     specifying, 2-151  
 SEND command, 2-223  
 SET BLOCKSIZE command,  
     2-225  
 SET BREAK command, 2-226  
 SET CDR command, 2-229  
 SET CPU command, 2-230  
 SET DEFAULT BUFFERS command,  
     2-232  
 SET DEFAULT PROTECTION  
     command, 2-233  
 SET DENSITY command, 2-235  
 SET DSKFUL command, 2-236  
 SET DSKPRI command, 2-237  
 SET FLOATING POINT command,  
     2-238  
 SET HOST command, 2-239  
 SET HPQ command, 2-241  
 SET PHYSICAL command, 2-243  
 SET RETRY command, 2-245  
 SET SPOOL command, 2-247  
 SET TERMINAL command, 2-249  
 SET TIME command, 2-250  
 SET TTY command, 2-252  
 SET VIRTUAL LIMIT command,  
     2-260  
 SET WATCH command, 2-262  
 SETSRC program, 2-265  
 Setting,  
     break points, 2-226  
     disk priority, 2-237  
     tape blocksize, 2-225  
     tape density, 2-235  
     terminal characteristics,  
         2-252  
 SFD, 1-19, 1-25  
 SIXBIT/ASCII codes, G-1  
 SKIP command, 2-270  
 Soft errors, 2-245  
 Space,  
     disk, 2-206  
 Spacing tapes, 2-10  
 Specifying,  
     directories, 1-23  
     file names, C-3  
     files, 1-12, 1-22, 1-23  
     paths, 2-146  
     search lists, 2-151  
 Spooling devices, 2-150,  
     2-247  
 SSAVE command, 2-219  
 Standard,  
     compilers, 2-22  
     file extensions, C-4, D-1  
     system names, D-1  
 START command, 2-271  
 Starting,  
     batch jobs, 2-273  
     jobs, 1-1, 2-146  
     programs, 2-18, 2-42,  
         2-217  
 Statistics,  
     disk usage, 2-86, 2-150,  
         2-206  
     job, 2-262  
 Status,  
     system, 2-275  
 Stopping,  
     programs, 2-123  
     terminal output, 1-6  
 Storing data, 1-12  
 Strings,  
     command, 1-7  
 Sub-file directories, 1-19,  
     1-25, 2-38  
 Subjobs, 2-177  
 SUBMIT command, 2-273  
 Suppressing terminal output,  
     1-6  
 Switch format, 1-10  
 SWITCH.INI files, B-1  
 Switches,  
     assembler, C-5  
     command, 1-10  
     compiler, C-5  
     processor, C-6  
 Symbols,  
     comparing, 2-119  
     listing global, 2-119  
 Syntax,  
     command, 1-7  
 SYSTAT command, 2-275  
 System,  
     configuration, 2-174  
     defaults, 1-22  
     information, 2-275  
     libraries, D-7  
 System libraries, 1-18  
 System names,  
     standard, D-1  
 System programs,  
     running, 2-207  
 System schedule, 2-221  
 Tables,  
     GETTAB, 2-49  
     symbol, 2-119  
 Tape,  
     error reporting, 2-245  
     rewinding, 2-216  
 Tape blocksize,  
     setting, 2-225

INDEX (CONT.)

Tape density,  
     setting, 2-235  
 Tape drive controller,  
     2-245  
 Tapes,  
     accessing, 2-11  
     moving, 2-10, 2-270  
     spacing, 2-10  
     unloading, 2-287  
 TECO,  
     running, 2-157  
 TECO command, 2-281  
 Temporary,  
     files, F-1  
     jobs, 2-7  
 Temporary switches, 1-11  
 Terminal,  
     characteristics, 2-128,  
         2-252  
     output, 2-286  
 Terminal characteristics,  
     setting, 2-252  
 Terminal number, 2-182  
 Terminal output,  
     continuing, 1-6  
     stopping, 1-6  
     suppressing, 1-6  
 Terminal types, 2-252  
 Terminals,  
     changing, 2-211  
     detaching, 2-69  
 Terminating,  
     commands, 1-2, 1-8  
     I/O, 2-21, 2-113  
     jobs, 2-211  
 TIME command, 2-282  
 Time limits,  
     CPU, 2-250  
 Timesharing, 1-1  
 Topology,  
     network, 2-171  
 TOPS-10 monitor, 1-1  
 TPUNCH command, 2-284  
  
 Transferring,  
     files, 2-31  
 TYPE command, 2-286  
 Type-ahead, 1-7  
 Typing errors,  
     correcting, 1-4, 1-5  
  
 UFD, 1-19, 1-25  
 UNLOAD command, 2-287  
 Unloading tapes, 2-287  
 Usage statistics,  
     disk, 2-86, 2-150, 2-206  
 User level,  
     entering, 2-30, 2-124  
 User-defined device names,  
     1-16  
 User-file directories, 1-19,  
     1-25, 2-38  
 Users,  
     authorizing, 1-1  
 USESTAT command, 2-288  
  
 VERSION command, 2-289  
 Virtual page limit, 2-152  
     current, 2-260  
  
 Warning messages, 3-2  
 WHERE command, 2-292  
 Wildcard characters, 1-22  
 Words,  
     deleting, 1-4  
 Writing core image files,  
     2-47  
  
 ZERO command, 2-294



-Do Not Tear - Fold Here and Tape-

**digital**



No Postage  
Necessary  
if Mailed in the  
United States

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

**SOFTWARE PUBLICATIONS**  
200 FOREST STREET MR1-2/E37  
MARLBOROUGH, MASSACHUSETTS 01752



-Do Not Tear - Fold Here and Tape-

Cut Along Dotted Line