TOPS–10
Operating System Commands Manual

AA–0916F–TB

October 1988

This manual contains descriptions of the TOPS–10 monitor commands, their formats and their usage. This manual replaces the *TOPS–10 Operating System Commands Manual*, order number AA–0916E–TB.

Operating System:	TOPS–10 Version 7.04
Software:	GALAXY Version 5.1

digital equipment corporation, maynard, massachusetts

First Printing, July 1975
Revised, August 1977
Revised, March 1978
Revised, August 1980
Updated, July 1982
Updated, February 1984
Revised, April 1986
Revised, October 1988

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 be used or copied only 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 Equipment Corporation or its affiliated companies.

The Reader's Comments form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

CI	DECtape	LA50	SITGO-10
DDCMP	DECUS	LN01	TOPS-10
DEC	DECwriter	LN03	TOPS-20
DECmail	DELNI	MASSBUS	TOPS-20AN
DECnet	DELUA	PDP	UNIBUS
DECnet-VAX	HSC	PDP-11/24	UETP
DECserver	HSC-50	PrintServer	VAX
DECserver 100	KA10	PrintServer 40	VAX/VMS
DECserver 200	KI	Q-bus	VT50
DECsystem-10	KL10	ReGIS	
DECSYSTEM-20	KS10	RSX	

digital™

CONTENTS

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 Volumes 1 and 2, the DECsystem-10 MACRO Assembler Reference Manual, and the DEC10/20 Processor 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 Introduction to DECsystem-10.

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, lists all the commands alphabetically, and describes each command in detail. Examples accompany the command descriptions.

The appendixes contain additional reference material.

CONVENTIONS USED IN THIS MANUAL

The following conventions are used in this manual:

Convention	Meaning
addr	Program or location address
arg	An argument to a command
c(addr)	The contents of an octal address
core	Refers to main (processor) memory
CTRL/x	A control character. Control characters are explained in Section 1.6.
date	A date in the form of dd-mm-yy. For example, 22-11-93 represents November 22, 1993.
date-time	The date and time in the standard format. (Refer to Section 1.8.3.)
density	The density of a magnetic tape
dev: device-name	Any logical or physical device name. A colon (:) must be included when a device name is part of a file specification. Refer to Section 1.9.1.
[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 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]. Refer to Section 1.9.
fs	File structure
h	High, referring to a job's high segment
hh:mm:ss	The time of day using a 24-hour clock (0 through 23), where hh is hours, mm is minutes, and ss is seconds
id	An identifier
job	A job number
jobname	The name of the job.
key	A keyword to a command argument.
l	Low, referring to a job's low segment
letter	An alphabetic character
lh	The left half of an octal word
line-number	Refers 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, such as 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] [ppn]	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 of a file
str	A file structure name
/switch	A command line modifier
time	Refers to a time of day
UFD	A user-file directory
v	Version number
x	A numeric or text variable
xx	A node number in a device name
$	The symbol printed on your terminal when you press the ESCape (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 (RUBOUT) key
<RET>	Indicates when you should press the RETURN key

<ESC>	Indicates when you should press the ESCape (ALTmode) key
|	Vertical bars (|) in the left margin denote changes or additions to the TOPS-10 Operating System since the last printing of this manual
o	Bullets (o) in the left margin denote deletions from the TOPS-10 Operating System since the last printing of this manual

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:

Introduction to DECsystem-10

Getting Started with DECsystem-10

TOPS-10/TOPS-20 Batch Reference Manual

TOPS-10 User Utilities Manual

TOPS-10 Monitor Calls Manual, Vols. 1 and 2

DECsystem-10 MACRO Assembler Reference Manual

DEC10/20 Processor Reference Manual

TOPS-10 Operator's Guide

TOPS-10 Software Installation Guide

TOPS-10 LINK Programmer's Reference Manual

TOPS-10/TOPS-20 FORTRAN Language Manual

TOPS-10/TOPS-20 DECmail/MS Manual

TOPS-10 DDT Manual

TOPS-10 MAKLIB User's Guide

VT52 Owner's Manual

LN01 Programmer's Reference Manual

CHAPTER 1

INTRODUCTION

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

1.1 JOBS

TOPS-10 is a timesharing 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. (The SUBMIT command is discussed in Chapter 2.)

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

1.2 CONNECTING TO THE SYSTEM

Only authorized users have access to TOPS-10. Your system administrator provides each authorized user with a user name, a project programmer number (PPN), and a password. These identify all users and their corresponding areas on file structures. When you specify a directory area, your project-programmer number identifies you. A comma separates the project and programmer numbers, and the entire PPN is 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 thirty-nine characters long and is used when you log in to the system and when you attach to another job. (Refer to the LOGIN and ATTACH commands, Chapter 2.) To maintain password secrecy, the monitor does not echo your password on the terminal. When you are using a terminal with local copy, a mask is printed when you type your password, making the password unreadable.

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 on 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. Section 1.4. discusses interactive levels.

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 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 1.8.)

Section 1.6 contains descriptions of the control characters and special keys you use when typing commands.

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 1.6.1.)

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 CONTEXTS

A **core image** is what a job's portion of memory contains at any given time. The core image, as well as information about the monitor's state with respect to your job, constitute a **context**. When you initialize a program, your context usually changes. For instance, running the DIRECT program creates a core image with the DIRECT program in it. Exiting DIRECT and then running SYSTAT, for example, destroys what you had in memory (the DIRECT program) and loads data pertinent to SYSTAT.

Three commands allow you to display information about contexts, and manipulate them in various ways. These commands are CONTEXT, PUSH, and POP. CONTEXT and PUSH create additional contexts. POP returns you to a superior context. When you work with multiple contexts, at least one context is preserved while you work with the current context.

You can work with multiple contexts by creating parallel contexts, or creating inferior contexts. The default maximum number of contexts, including current, parallel, and inferior, is four. The number of contexts in use at any moment is shown by the CONTEXT command. You can also use the CONTEXT command to create parallel contexts. You might create a parallel context, run a frequently accessed program in it, and then exit, returning to your previous context. Both contexts now exist simultaneously. When you need to use the program in the created context, simply switch to that context. You will not have to wait for it to reinitialize. The full description of the CONTEXT command in Chapter 2 gives an example of this procedure, including how to create and access a parallel context.

The PUSH command allows you to create an inferior context. When you return to your previous task in a superior context, the inferior context is deleted. You return to the superior context using the POP command. An inferior context could be used in the following situation: If you were in the process of completing a task, and needed to see some HELP text, you could create an inferior context, read the HELP text you needed, destroy the inferior context and return to the unchanged superior context. The system automatically creates a new context for certain commands. The system manager can set the system to create a new context when any monitor command is issued.

1.6 SPECIAL CHARACTERS

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

These special characters are described in the following sections.

1.6.1 CTRL/C - Interrupt

When you type a CTRL/C (control-C), the monitor 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

.
```

1.6.2 DELETE Key

When you press the DELETE (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 monitor 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>

.
```

1.6.3 CTRL/W - Delete Word

When you type CTRL/W (control-W), the monitor 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.

Example

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

```
.SET TTY TYPE L036\630L\LA36<RET>

.
```

1.6.4 CTRL/U - Delete Line

When you type CTRL/U (control-U), the monitor 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>
```

1.6.5 CTRL/R - Reprint Line

When you type CTRL/R (control-R), the monitor reprints the current input line. If you type a line incorrectly, then make corrections using the DELETE key, the monitor 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 N0<DEL>O FILE<DEL>L
```

The line appears as:

```
.SET TTQ\Q\Y N0\0\O FILE\E\L
```

If you then press CTRL/R:

```
.SET TTQ\Q\Y N0\0\O FILE\E\L ^R
```

The monitor prints:

```
SET TTY NO FILL
```

When you type CTRL/R, the monitor 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.

If a program such as your text editor uses CTRL/R for another purpose, you can disable the CTRL/R function using the SET TTY RTCOMP command. (See Chapter 2 for a complete description of this command.)

1.6.6 CTRL/O - Cancel Output

When you type CTRL/O (control-O), the monitor cancels 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 monitor to finish printing the message, you can stop the monitor 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 monitor continues executing your program, but does not print any output on your terminal. The monitor 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.

1.6.7 CTRL/S - Hold Output

When you type CTRL/S (control-S), the monitor holds 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.

1.6.8 CTRL/Q - Resume Output

When you type CTRL/Q (control-Q), the monitor 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.

1.6.9 CTRL/T - Job Status

When you type CTRL/T (control-T), the monitor prints status information pertaining to your job on your terminal. CTRL/T does not echo on your terminal. There are 10 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.)

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 current context number (for example, Ctx:1.)

8. 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.

9. The program counter, which is the address of the current instruction. (For example, PC:400275.)

10. 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.)

If a system program, such as your text editor uses CTRL/T for another purpose, you can disable the CTRL/T function using the SET TTY RTCOMP command. (See Chapter 2 for a complete description of this command.)

Example

```
DAY: :05:43 RUN:0.48 RD:75 WR:8 SOS 12+19P Ctx:1 PC:400275
```

1.7 TYPE-AHEAD CAPABILITY

Type-ahead allows you to type another command without waiting for the monitor 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 monitor prints after completing the first operation. If an error occurs during the first operation, the characters you typed ahead are ignored.

1.8 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
```

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 monitor ignores it. You need only type enough characters to uniquely identify the command.

It is sometimes possible to abbreviate a command by typing characters fewer than would make the command unique. This is not encouraged, however, because uniqueness of a command may vary from monitor release to monitor release. For this reason, you should use the whole command in a batch control file.

1.8.1 Command Termination

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

```
.DIRECT<RET>
```

1.8.2 Line Continuation

You can continue a command line to some system programs, such as DIRECT and QUEUE, by placing a hyphen (-) as the last nonblank, noncomment character 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.

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

1.8.3.1 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.

1.8.3.2 Absolute Date-Time Arguments - An absolute argument specifies a particular date or time. The format of an absolute argument is one of the following:

- weekday:hours:minutes:seconds
- date:hours:minutes:seconds
- keyword: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
```

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, 90, and 1990 will all be interpreted as the year 1990.

keyword is one of the following options:

LOGIN (time of login)

NOON

MIDNIGHT

hours is based on a 24-hour clock (0 through 23) and is required if you omitted the weekday, date, or keyword, or if you specify minutes.

minutes is optional unless you specify seconds.

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-3-85 is since July 3, 1985 at midnight
```

1.8.4 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
```

1.8.4.1 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.

1.8.4.2 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.

1.8.5 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 <RET>
```

1.9 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 monitor 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 1.9.1. Always type a colon following the device name. An example of a device name is DSKC:.

A **file name** is one to six alphanumeric characters. The monitor ignores all characters past the sixth. File names are discussed in Section 1.9.2. 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 1.9.3. 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 1.9.4. An example of a full file specification is DSKC:MYFILE.TXT[21,589]<055>.

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>
```

1.9.1 Device Names

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

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

The monitor 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.

To reference 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 1.9.1.1 through 1.9.1.5.

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.

1.9.1.1 Generic Device Names - The most general type of device name is the generic device name. When you specify a generic name, the monitor selects a free unit of the device type specified. When your computer is in a network environment, the monitor chooses the device from those devices at your location, or, if none are available, the monitor 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 1.12.)

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:
	RPx:	RP:
Fixed head	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 monitor 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 monitor always selects your terminal (assuming that these are not logical names).

1.9.1.2 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, MTA1: indicates magnetic tape unit 1 on
o controller A.

1.9.1.3 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 four 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.

File structures are illustrated and explained in detail in the TOPS-10 Monitor Calls Manual Vol. 1.

1.9.1.4 Logical Device Names - You can assign a logical name to a physical device. The monitor 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 monitor 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 MTA3: (magnetic tape unit 3 on controller A), all output to generic DSK will go to the magnetic tape.

Some names are recognized by the monitor as special system-defined logical names that the monitor 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 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**.

Example

You request a DECtape drive with the logical name ABC:

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

The monitor gives you unrestricted device DTA0: (DECtape number 0 on controller A). You then mount a DECtape on drive DTA0:

```
DTA0 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, PTP0 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 DTA0: to logical device ABC: (physical device PTP0:).

```
*ABC:=DTA0:FILEA.EXT
```

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 (which you already have assigned to you) be assigned the logical name DEF:

```
.ASSIGN DTA0: DEF:
```

DECtape unit 0 takes on the logical name DEF:.

```
DTA0 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 DTA0: (DECtape drive 0) again and request that the logical DEF: be changed to XYZ:

```
.ASSIGN DTA0: XYZ:
```

The system disassociates the logical name DEF: from DTA0:. DECtape unit 0 takes on the logical name XYZ:.

```
DTA0 ASSIGNED
```

1.9.1.5 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. Appendix D contains a complete list of the ersatz device names used by TOPS-10.

1.9.2 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.

1.9.3 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]

A Sub-File Directory (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 1.14.

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

1.9.4 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:

```
<0      5       7>
 |      |       |
 |      |       |
 |      |       Protection code for all other users.
 |      |
 |      |
 |      Protection code for all members of file owner's
 |      project.
 |
 |
Protection code for the file owner.
```

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*, 6*	You can execute, read, or change the protection code of the file.
5*	You have unlimited access to the file, except for renaming it.
4*	You have unlimited access to the file.
3	You can execute, read, or change the protection code of the file.
2	You have unlimited access to the file, except for renaming it.
1, 0	You have unlimited access.

* The File Daemon is called on a protection failure on this file

Table 1-4: Protection Codes for Fields 2 and 3

Project-members and Other User Protections	
Code	**Access Privileges**
7	The user cannot access the file.
6	The user can only execute the file.
5	The user can execute or read the file.
4	The user can execute, read, or append to the file.
3	The user can execute, read, append to, or update the file.
2	The user can execute, read, append to, update, and write to the file.
1	The user can execute, read, append to, update, write to, and rename the file.
0	Unlimited access, including changing the protection code of the file.

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 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.
- The standard TOPS-10 default protection code that is defined by your installation (usually <055> or <057>).

1.9.5 File Daemon

The File Daemon, a system program, may determine the protection status of files with a file protection code of 4, 5, 6, or 7 in the owner's protection field. (Protection codes are discussed in Section 1.9.4.) When you set such a protection, a protection violation causes the File Daemon to access a file 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 cannot access the file. The File Daemon and the file ACCESS.USR are described in the FILDAE Specification in the TOPS-10 Notebook Set.

1.10 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 1.12 and 1.14.

1.11 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.

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 DIRECTORY and QUEUE programs recognize a number sign (#) in the file specification to indicate that SIXBIT octal code follows. For example,

```
.DIR #640000000000.
```

gives the same directory listing as

```
.DIR T.
```

because 120000000000 is the 36-bit left-justified SIXBIT code for the character "T". This function is useful for file names which contain special characters. Although the SIXBIT code for an asterisk is 120000000000, the command line

```
.DIR #120000000000.
```

would not give the same directory listing as

```
.DIR *.
```

because the first command would list only a file literally named *. The second command would interpret the * as a wildcard and would list all files without extensions.

You cannot mix SIXBIT code and regular characters within the file name or within the file extension. However, you can combine a SIXBIT file name with a regular file extension and vice versa.

Appendix G contains a chart of SIXBIT and ASCII character codes.

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 the SETSRC Program description in the TOPS-10 User Utilities Manual.)
[,,SUB1,SUB2]	The sub-file directory SUB2 under the sub-file directory SUB1 in your UFD

1.12 SEARCH LISTS

A search list is a list of file structures listed in the order that the operating system will search through them for a file. There are two types: the system search list and a job search list.

The system search list, designated by the ersatz device name SSL:, contains all files pertinent to the daily operation of the system. It is the same for every job on the system. Only the system administrator can define and change this list. However, you may display the system search list using the SETSRC program.

Some of the types of files kept in the system search list file structures include: accounting files, help and documentation files, compilers, and system programs.

The job search list, designated by the ersatz device name DSK:, lists structures that contain user files for an individual job. The system administrator creates the default job search list for each new user account. You can display, add, remove, and rearrange the file structures in your job search list using the SETSRC program. (Refer to the TOPS-10 User Utilities Manual for more information about SETSRC.) The MOUNT and DISMOUNT commands also modify your job search list. (MOUNT and DISMOUNT are described in Chapter 2.)

For example, in the following command line

```
.MAKE TEST.LIS
```

the system uses the system search list to find the program (TECO) invoked by the MAKE command. The system uses your job search list to find your user file TEST.LIS.

The format of the job 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 indicates how the file can be accessed (read only, for example).

The job search list has two parts, the active search list and the passive search list. The active search list is the list of file structures on the left side of the fence. The system searches each of these file structures from left to right.

The passive search list is to the right of the fence. It is composed of the file structures that you removed from the active search list using the SETSRC program. Therefore, when you first log in, you have no structures in your passive search list. The file structures in the passive search list are not searched by the the monitor, but are used to compute disk usage when you log off the system.

1.13 LIBRARIES

The system libraries contain compilers, system programs and other important files. Wherever a device can be specified, an ersatz device may be specified instead. An ersatz device is a monitor-defined logical name for a directory specification. There are several special ersatz devices defined for some libraries. There are three separate PPNs, [1,3], [1,4], and [1,5], where system library files can be stored. The standard version of a file or program is usually found on SSL:[1,4].

The newest version of a file, sometimes in an experimental or untested state, can sometimes be found in the [1,5] system library. NEW: is a special ersatz device. When the operating system looks for a file on NEW: it will first look on SSL:[1,5]. If the file is not found there, however, it will next look on SSL:[1,4] for the same file. Likewise, an outdated version of a file might be found on OLD:, which searches SSL:[1,3] before SSL:[1,4].

The current system library for the job is called SYS:. By default SYS: is SSL:[1,4] or the standard system library. You may, however, change the default definition of SYS: by specifying the /NEW switch to LOGIN or to the SETSRC program. This changes the definition of SYS: to be NEW:. In other words, the system looks on SSL:[1,5] before looking on SSL:[1,4]. When you use the R command, the program comes from SYS:.

LIB: is the job's library directory. If you define LIB:, any file the system cannot find on DSK: will be searched for in the user-file directory (UFD) defined in LIB:. LIB: may be set by the SETSRC program.

1.14 DIRECTORY PATHS

A directory path is an ordered list of directory names, starting with a user-file directory, that uniquely identifies a directory. Directory names are discussed in Section 1.9.3.

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

You can change your default directory path using the SETSRC program. See the TOPS-10 User Utilities Manual for more information about SETSRC.

Sub-file directories allow you to organize your files, and to access them in sets. Any directory acts as an index to a set of files. An SFD is pointed to by a file in the UFD, or by a higher-level SFD. You create SFDs with the CREDIR program, 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 five. (See the TOPS-10 User Utilities Manual for more information about the CREDIR program.) Nested directories form a directory tree structure, which is illustrated in the TOPS-10 Monitor Calls Manual.

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<RET>

NUMB        TST   0   <055>   dd-mmm-yy        DSKC:   [27,5434]
109         TST   5   <055>   dd-mmm-yy
FILL        TST   1   <055>   dd-mmm-yy
PAY3        TST   1   <055>   dd-mmm-yy
   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
```

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.*<RET>

TEST        FOR   1   <055>   dd-mmm-yy        DSKC: [27,5434]
TEST        SFD   1   <775>   dd-mmm-yy
   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<RET>
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<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>   dd-mmm-yy     DSKC: [27,5434,TEST]
109         TST   5   <055>   dd-mmm-yy
FILL        TST   1   <055>   dd-mmm-yy
PAY3        TST   1   <055>   dd-mmm-yy
  TOTAL OF 7 BLOCKS IN 4 FILES ON DSKC: [27,5434,TEST]
```

1.15 USER-DEFINABLE COMMANDS

User-definable commands are available at both the system programming level and at the monitor command level for the timesharing user.

As a timesharing user, you use the DECLARE command to define a command to run a specified program for your job. This can be done during a timesharing session, or you can include the DECLARE command in a SWITCH.INI file. Any program can be invoked with a user-defined command.

For example, you can define COMPAR as a user-defined command to execute the FILCOM program as follows:

```
.DECLARE  COMPAR=SYS:FILCOM.EXE<RET>
```

You can also use DECLARE to display your job's command list. The commands in the following list were defined previously.

```
.DECLARE/LIST<RET>

COMPAR   LOOK   EDIT   DISKSP
```

Use the COMPAR command. Use CTRL/T to display the current state of your job, including the name of the program that is running. Note that, although CTRL/T is shown here, it does not echo on your terminal. CTRL/Z is used to exit FILCOM and return to the monitor.

```
.COMPAR<RET>
*<CTRL/T>

Day: 14.83 Run: 0.14 R2.5 Wr:6 FILCOM 3+4P TI PC:401570
Input wait for TTY70:
^Z
.
```

For more information on user-definable commands, see the DECLARE command.

1.16 PROGRAMMING THE LN01 LASER PRINTER

This section describes the use of escape and control sequences, which you use to direct printing on the LN01 laser printer. Font loading, which is one application of control sequences, and font management are also discussed. For a detailed discussion of programming on the LN01, refer to the LN01 Programmer's Reference Manual.

1.16.1 LN01 ESCAPE AND CONTROL SEQUENCES

You can use escape and control sequences to specify tab stops or margins, change fonts within a file, or indicate the text's orientation, for instance. ESCape and control sequences are typed in as part of a file you want to print on the LN01 laser printer. Every time the printer encounters the characters that introduce an escape or control sequence, it regards the next few characters as instructions, until it encounters a final character (described below). However, you must specify the /PRINT:GRAPHICS switch to the PRINT or QUEUE command in order for the sequences to be interpreted. Otherwise, these characters will be printed as part of the file.

For a complete description of all valid escape and control sequences, please refer to the LN01 Programmer's Reference Manual.

NOTE

You use the ESC control character as the first character of all escape and control sequences. When referred to here, it means that you actually press the ESC key. Since the ESC key is interpreted as a function by most text editors, consult the reference manual to find out how your editor can actually print the ESC control character. The examples in this section use a dollar sign ($) to show that the ESC control character has been pressed.

1.16.1.1 ESCape Sequences - An escape sequence can have three sections: an introducer, intermediate characters, and a final character. You must include the escape sequence introducer, which is the ESC control character, and the final character; the intermediate characters are optional. Intermediate characters define the interpretation of the sequence. The final character indicates the end of a string, and is defined by the function requested in the sequence.

The following sequence shows how to reset a printer to a known state:

```
ESC control character
 |    Final character
 |       |
 |       |
 $       c
```

The characters are spaced here for clarity only. The sequence must appear as $c to actually be interpreted by the LN01. Case is significant when final characters are alphabetic.

1.16.1.2 Control Sequences - The format for control sequences is similar to that of escape sequences. However, control sequences begin with an ESC control character and a left square bracket. Also, one or more parameters are allowed before the intermediate character(s). Parameters are separated by the use of a semicolon (;).

Numeric parameters that designate length (such as specifying 5 inches) are given in terms of points or decipoints. A point is a unit of length equal to 1/72 of an inch. A decipoint, then, is a length of 1/720 of an inch. In these units, 5 inches equals 360 points or 3600 decipoints.

A sequence using length in points is below. In this example, the left margin is set to 5 inches (360 points), and the right to 8 inches (576 points).

```
$[          3 6 0 ; 5 7 6          s
|           |                      |
|           |                      Final character
|           |
|           Numeric parameter
|
Control sequence characters
```

In this example, a control sequence is used to clear all horizontal tab stops.

```
$[      3       g
|       |       |
|       |       Final character
|       Numeric parameter
Control sequence characters
```

Remember, as with escape sequences, there should be no blank spaces between the characters. (They would appear in the file as "$[3g".)

1.16.2 Font Management

The LPTSPL program supports font management for the LN01 laser printer by means of a font handler. LPTSPL requires the LPFONT.INI file to automatically process fonts by name. LPFONT.INI resides on SSL:[5,36,LN01]. LPFONT.INI maps the font names to the files in which the actual fonts are kept. Font files must be kept in the same area.

LPFONT.INI has the format:

```
filenm = fontnm1<TAB>fontnm2<TAB>...<TAB>fontnmN
```

Where: **filenm** is the six character filename of the font.

fontnm is an alphanumeric string from 1 to 30 characters long, representing a fontname. Each fontname must be separated by at least one tab character, and the line must be ended with a carriage return/line feed.

1.16.3 Loading Fonts

When you want to print a file using non-standard fonts, you have two options. You may use the either /FONT:fontnm or /PRINT:GRAPHICS switches to the PRINT or QUEUE commands.

Use the /FONT:fontnm switch when you want an entire file printed in the same font. LPTSPL loads the font you request, and the LN01 prints your file in the font you specified.

The /PRINT:GRAPHICS switch allows you to dynamically load fonts, so that you can print a file in several different fonts. The Load Character Set (LCS) and Assign Character Set (ACS) sequences are used to prepare the fonts for printing. If you are using fonts which reside on the ersatz device FNT: ([5,36,LN01]), you do not need to explicitly specify the LCS sequence; LPTSPL loads the character sets for you. You will always need to place an ACS sequence at the beginning of your file. The ACS sequence assigns a font name to a number. You refer to this number immediately before the text you want printed in that font. The text continues printing in that font until you specify a new font. For example, if you assigned 13 to represent a Times Bold font, and 14 to represent Times Italic, you could print text in those fonts using the following:

```
$[13mPrint in Times Bold and $[14mnow switch to Italic
```

In the above example, the ESC/square bracket combination indicates a control sequence is about to follow, the 13 and 14 represent the fonts, and the letter m is the final character. For more information on font programming on the LN01, refer to the LN01 Programmer's Reference Manual.

CHAPTER 2

SYSTEM COMMANDS

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

2.1 FUNCTIONAL GROUPS OF COMMANDS

In Section 2.1, the TOPS-10 commands 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 in Section 2.2 before using the command.

2.1.1 Job-Control Commands

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	CONTEXT	DECLARE
DETACH	DISABLE	ENABLE
KJOB	LOGIN	PASSWORD
POP	PUSH	REATTACH
SESSION		

2.1.2 Information Commands

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

ACCOUNT	ALLOCATE	CONTEXT	CORE
CPUNCH	DAYTIME	DIRECTORY	DSK
HELP	INITIA	MOUNT	NETWORK
NODE	PJOB	PLOT	PRINT
PUNCH	QUEUE	RESOURCES	SCHEDULE
SET WATCH	SHOW ALLOCATION	SHOW QUEUES	SUBMIT
SYSTAT	TPUNCH	USESTAT	VERSION
WHERE			

2.1.3 Terminal-Control Commands

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 communicate with a user at another terminal. The commands that allow you to do this are:

MAIL	PLEASE	SEND

2.1.5 File-Handling Commands

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 are:

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

2.1.6 Device-Handling Commands

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

ALLOCATE	ASSIGN	BACKSPACE
CANCEL	CLOSE	CPUNCH
DEALLOCATE	DEASSIGN	DISMOUNT
EOF	FINISH	LABEL
LIST	LOCATE	MOUNT
PLOT	PRINT	PUNCH
QUEUE	REASSIGN	REWIND
SET BLOCKSIZE	SET CDR	SET DEFER
SET DENSITY	SET FORMAT	SET RETRY
SET SPOOL	SKIP	TPUNCH
UNLOAD		

2.1.7 Program-Preparation Commands

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 are:

CLOSE	COMPILE	CREF
DDT	DEBUG	DEPOSIT
EOF	EXAMINE	FUDGE
LOAD	MAKE	MERGE
SET BREAK	SET DDT BREAKPOINT	TECO

2.1.8 Program-Control Commands

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	CONTEXT	CONTINUE
CORE	CSTART	EXECUTE
GET	HALT (CTRL/C)	JCONTINUE
LOAD	MERGE	POP
PUSH	R	REENTER
RUN	SAVE	SSAVE
SET BREAK	SET CPU	SET DEFAULT BIGBUF
SET DEFAULT BUFFERS	SET DSKFUL	SET DSKPRI
SET HPQ	SET PHYSICAL	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

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 or @	Executes a MIC command file.
BACKTO GOTO	Specifies a label at which MIC processing is to resume within the command file.
COJOB	Creates a COJOB.
ERROR/ NOERROR	Specifies an error condition character.
IF	Conditionally processes a monitor command.
LET	Changes the values of user parameters.
MIC	Passes a subcommand to MIC.
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

ACCOUNT Command

Function

If the monitor is running usage file accounting software, the ACCOUNT command prints the account you are logged in under (set by a LOGIN or SESSION command) on your terminal. If you do not have an account, the monitor issues only a carriage-return.

Format

ACCOUNT

After you press RETURN, the monitor prints the account you are logged in under.

Characteristics

Leaves your terminal in monitor mode.

Requires LOGIN.

Examples

1. This example indicates that you do not have an account string.

   .ACCOUNT<RET>

   .

2. The ACCOUNT command prints the string TS547 as the name of your account.

   .ACCOUNT <RET>

   TS547

   .

ALLOCATE Command

Function

The ALLOCATE command informs the system that you will need a resource. A resource is a tape or disk unit. With ALLOCATE, you can reserve a resource from the system and assign a logical name to the resource you will use.

An allocated resource can contain all the specifications necessary when the resource is mounted. The switches that you specify with the ALLOCATE command will be retained and applied when you use the MOUNT command to mount the same resource.

The ALLOCATE command allocates a resource, and the SHOW ALLOCATION display will show such a resource as both allocated and mounted. If you use the ALLOCATE command to explicitly allocate a resource, you will be granted extended ownership over the resource. For example, if a resource that you have explicitly allocated and mounted goes off-line unexpectedly (as when the operator dismounts it), your mount request for that resource will be automatically requeued, and the resource will be mounted for your job when it comes on line.

Format

```
ALLOCATE resource:log-name/switch/switch...
```

Where: **resource** is the name of the resource that is to be allocated. The colon (:) in the resource name is optional. The resource name is one of the following:

- A disk structure or volume set name, such as DSKB:.
- A tape volume set name and volume identifiers, such as PAY-WK(PM34,PM35) where PAY-WK is the volume set name, and (PM34,PM35) is a list of the volumes in the volume set.
- A tape volume identifier of a single-tape volume set.
- The logical name previously associated with a resource.
- A physical device name.

Note that a tape allocation request requires a volume identification. If you do not include the volume set name followed by volume identifiers, you must supply the /VOLID switch.

log-name is the logical name you can assign to the resource that you will use. The logical name can be up to 6 alphanumeric characters. A tape volume set must always have a logical name. If you do not specify a logical name for a tape volume set, the system defaults to the first six characters of the volume set name, or up to the first non-alphanumeric character of the volume set name. A disk volume set does not require a logical-name.

To allocate more than one volume set, separate the volume set identifications with commas.

/switches are always preceded by a slash. Some switches can be used for any kind of volume set; others are restricted to either tape or disk volume sets only.

NOTE

The logical name and switches that you specify in the ALLOCATE command string are saved by the system, and are applied when you MOUNT the volume set.

You can obtain a list of the resources that are allocated and mounted for your job by typing ALLOCATE with no arguments or switches. The output is the same as the output from the SHOW ALLOCATION command.

The following is a list of the switches you can use with ALLOCATE. The center column lists the kind of resource(s) the switch applies to.

Switch	Device	Function
/ACTIVE	Disk	Requests that the volume set be placed in your job's active search list when the structure is mounted. (See SETSRC in the TOPS-10 User Utilities Manual). The structure will become part of the list that the system automatically uses to search for a file. This is the default. Complement to /PASSIVE.
/CHECK	Tape Disk	Prints a list of all the allocation requests for your job. (Same as ALLOCATE with no arguments or switches.)
/CREATE	Disk	Allows files to be created on this structure. This switch is the complement to /NOCREATE and implies the /ACTIVE switch.
/DENSITY:n-BPI	Tape	Specifies the recording density (bits-per-inch) of the volume set. The density (n) can be: 200, 556, 800, 1600, or 6250. The -BPI portion of the value is optional.
/DISK	Disk	Identifies the volume set as a disk volume set.
/EXCLUSIVE	Disk	Ensures that you will have exclusive access to the resource. No other users will be allowed to access the resource. You must have the same project number as the owner of the volume set.
/HELP	Tape Disk	Prints a brief description of the command.

/LABEL-TYPE:arg Tape

Specifies the kind of label processing to be used and indicates the label status of the tape. The arguments and their meanings are:

ANSI	The label is formatted according to ANSI standards.
BLP BYPASS	The tape may or may not have labels, but it is treated as if it were unlabeled. Only privileged users can use this switch.
EBCDIC IBM	The label is formatted in EBCDIC.
USER-EOT	The tape does not have standard labels. However, you will be informed at the end of tape. Volume switching (for multivolume tape volume sets) will not be performed automatically. Therefore, the user program is responsible for mounting subsequent tapes. This is useful for programs that create unique labels, such as BACKUP.
UNLABELED NONE NOLABELS	The tape is not labeled. When a new tape is mounted from the same volume set, you will not be notified. The switching of tapes in the volume set will be handled automatically by the system. You will not be informed when the end of tape is reached.

/NEW-VOLUME-SET Tape

Specifies that a new volume set is going to be created. The operator will choose tapes for your job from a pool of available tapes. This switch implies /WRITE-ENABLE.

/NOCREATE Disk

Prevents the creation of files on this volume set, unless you specify the volume set when you write the file. This switch is the complement to /CREATE and it implies /ACTIVE.

/NONOTIFY Tape Disk

Does not inform you when the resource is mounted or dismounted.

/NOTIFY	Tape Disk	Sets the system to inform you when the resource is mounted or dismounted. The system sends a message to your terminal when any of the following occurs: o The resource is physically mounted. o The resource is dismounted by the operator without a request by your job. o Another volume in a tape volume set is mounted. o The disk structure is locked or unlocked by the operator.
/PASSIVE	Disk	Requests that the structure be placed in your job's passive search list. (Refer to SETSRC program.) The system will not search for files in the structure unless you specify the structure name in the file specification. This switch is the complement to /ACTIVE.
/QUOTA	Disk	Removed.
/READ-ONLY	Tape Disk	Specifies that you will not write on the volume set. Tape volume sets will be checked as they are mounted, to ensure that they are write-locked. This is the default for tape volume sets. On disk volume sets, the monitor will not update access dates. If you specify both /EXCLUSIVE and /READ-ONLY, the operator may write-lock the structure.
/REMARK:"text"	Tape Disk	Allows you to send a message to the operator identifying the volume to be mounted. The argument (text) can be up to 50 characters long. Use quotation marks around the text if it contains spaces or punctuation marks.
/SCRATCH	Tape	Instructs the operator to mount a scratch tape. By implication, a scratch tape will be returned to the system's pool of available tapes, after you are finished with it. This switch implies /WRITE-ENABLE.
/SHARABLE	Disk	Allows other users to access the resource. This is the default function. This switch is the complement to /EXCLUSIVE.
/TAPE	Tape	Specifies that the volume set is a tape volume set. This switch is required when the volume set has the same name as a catalogued disk volume set.

/TRACKS:n	Tape	Specifies the number of tracks on the tape. The value n can be 7 or 9.
/VOLID:volid	Tape	Identifies the volumes in a tape volume set. This switch can be used only if the volid-list was not specified previously. If the volume set is comprised of more than one volume, the volids should be separated by commas, and the volid-list should be enclosed in parentheses.
/WRITE-ENABLE	Tape Disk	Ensures that you can write on the volume set. For tape, the system checks each volume as it is mounted to be sure that it is write-enabled. This is the default for disk volume sets.

Associated Commands

MOUNT	Makes a device available to you.
DISMOUNT	Removes the specified volume set from your job's search list. Dismounts the volume set if you have exclusive access to it.
DEALLOCATE	Removes the specified resources(s) from your job's list of allocated resources.
SHOW ALLOCATION	Prints a list of the resources that are allocated and mounted for your job.
SHOW QUEUE	Prints a list of system queues.
CANCEL	Cancels a mount or queue request.

Characteristics

Runs the QUEUE program.

Destroys your core image.

Requires LOGIN.

Example

The following example shows the use of the ALLOCATE, DEALLOCATE, MOUNT, DISMOUNT, and SHOW ALLOCATION commands. The resources are reserved for a multivolume tape volume set with the ALLOCATE command. The name of the volume set is TAPE-SET, and it contains three volumes. The logical name TS is assigned to the tape set. The tape is write enabled, and it does not have standard labels.

```
.ALLOCATE TAPE-SET(VOL1,VOL2,VOL3):TS/WRITE-ENABLE/LABEL:NONE<RET>
[ALLOCATE REQUEST TS QUEUED, REQUEST #672]
```

A file structure named DSKR: is mounted for the job:

```
.MOUNT DSKR:<RET>
[MOUNT REQUEST DSKR QUEUED, REQUEST #673]
[STRUCTURE DSKR MOUNTED]
```

The job's resources are shown using the SHOW ALLOCATION command:

```
.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
   VOLUME SET           RESOURCES          TYPE        ALL   OWN
----------------      ----------------  ------------ ---   ---
---                   9 TK 800/1600     MAGTAPE UNIT  1     0
---                   RP06              DISK UNIT     2     2
---                   RP20              DISK UNIT     1     1
DSKB                  DSKB              STRUCTURE     1     1
DSKC                  DSKC              STRUCTURE     1     1
DSKR                  DSKR              STRUCTURE     1     1
TAPE-SET              VOL1              MAGTAPE VOL.  1     0
TAPE-SET              VOL2              MAGTAPE VOL.  1     0
TAPE-SET              VOL3              MAGTAPE VOL.  1     0
```

The tape set is mounted, and the resources are again displayed:

```
.MOUNT TS<RET>
[MOUNT REQUEST TS QUEUED, REQUEST #673]
[MAGTAPE TS MOUNTED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
   VOLUME SET           RESOURCES          TYPE        ALL   OWN
----------------      ----------------  ------------ ---   ---
---                   9TK 800/1600      MAGTAPE UNIT  1     1
---                   RP06              DISK UNIT     2     2
---                   RP20              DISK UNIT     1     1
DSKB                  DSKB              STRUCTURE     1     1
DSKC                  DSKC              STRUCTURE     1     1
DSKR                  DSKR              STRUCTURE     1     1
TAPE-SET              VOL1              MAGTAPE VOL.  1     1
TAPE-SET              VOL2              MAGTAPE VOL.  1     0
TAPE-SET              VOL3              MAGTAPE VOL.  1     0
```

After work is finished by accessing the tape set and the structure, the structure is dismounted. Because the structure was not explicitly allocated, it is automatically deallocated:

```
.DISMOUNT DSKR<RET>
[STRUCTURE DSKR DISMOUNTED]
```

The tape volume set is dismounted:

```
.DISMOUNT TS<RET>
[VOLUME SET TS DISMOUNTED]
```

The job's resources are displayed:

```
.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
   VOLUME SET            RESOURCES          TYPE        ALL  OWN
----------------      -----------------  ------------ ---  ---
---                   9 TK 800/1600      MAGTAPE UNIT  1    0
---                   RP06               DISK UNIT     1    1
---                   RP20               DISK UNIT     1    1
DSKB                  DSKB               STRUCTURE     1    1
DSKC                  DSKC               STRUCTURE     1    1
TAPE-SET              VOL1               MAGTAPE VOL.  1    0
TAPE-SET              VOL2               MAGTAPE VOL.  1    0
TAPE-SET              VOL3               MAGTAPE VOL.  1    0
```

At this point, the tape set can again be mounted, or it can be deallocated. The tape set is dismounted:

```
.DEALLOCATE TS<RET>
[VOLUME SET TS HAS BEEN DEALLOCATED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
   VOLUME SET            RESOURCES          TYPE        ALL  OWN
----------------      -----------------  ------------ ---  ---
---                   RP06               DISK UNIT     1    1
---                   RP20               DISK UNIT     1    1
DSKB                  DSKB               STRUCTURE     1    1
DSKC                  DSKC               STRUCTURE     1    1
```

.

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 1.9.1.) 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.

Format

ASSIGN node-id_device-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 is optional. If you omit the node-id, the node to which your terminal is connected is assumed.

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

device-name: is any physical device name or previously assigned logical name. (Device names are discussed in Section 1.9.1.)

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.

Characteristics

Leaves your job 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 . If you specify a restricted device, the monitor prints the following message:

?DEVICE NOT ASSIGNABLE

You can access restricted devices using the MOUNT command. The following message occurs when the device is assigned to another job:

?Already assigned to job n

Where: **n** is the number of the job to which the device is assigned.

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.

Formats

```
ATTACH
ATTACH job [ppn]/switch
ATTACH job/switch
ATTACH [ppn]/switch
ATTACH user/switch
```

Where: **job** is the job-number of the job to which your terminal is to be attached.

[ppn] 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.

user is the user-name associated with the desired job.

/switch is one of the following options:

/HELP:keyword	Prints the HELP text. Valid keywords are ARGUMENTS, SWITCHES, and TEXT. The ARGUMENTS keyword displays a list of valid switches and arguments. The SWITCHES keyword displays only a list of switches without detailed information. The TEXT keyword displays the full HELP text. TEXT is the default keyword. /HELP may be abbreviated to /H.
/MAILCHECK: YES or NO	If YES, checks for the existence of new mail.
/MONITOR	Leaves your terminal at monitor level.
/SETTTY: YES or NO	If YES, sets terminal parameters as specified with the /TERMINAL switch.

/TERMINAL:key	Defines the terminal characteristics. **key** is a keyword. The keywords for /TERMINAL are described after the switches.
/USER	Leaves your terminal at user level.

/Terminal Keywords

The /TERMINAL switch takes a list of parameters to specify terminal attributes. You can include multiple keywords for the /TERMINAL switch, in which case you must enclose them in parentheses and separate them with commas. Valid keywords are:

ALTMODE:yes-no	Do [not] convert ASCII 175 and 176 to ESCAPE (Altmode (ASCII 33)).
BLANKS:yes-no	Do [not] print blank lines.
CRLF:yes-no	Do [not] give a free CRLF at right margin.
DEFER:yes-no	Do [not] set deferred echo mode.
DISPLAY:yes-no	Terminal is [not] a display terminal.
ECHO:yes-no	Do [not] set terminal echo.
EIGHTBIT:yes-no	Do [not] set 8-bit mode.
ESCAPE:chr	Set <ESCAPE> translation character to chr.
FILL:dnum	Set filler class to dnum (0<=dnum<=3).
FORM:yes-no	Terminal does [not] have hardware form feeds.
GAG:yes-no	Do [not] accept SEND messages at user level.
LC:yes-no	Terminal does [not] have lowercase characters.
LENGTH:dnum	Set the terminal screen length to dnum.
LOCALCOPY:yes-no	Do [not] set terminal to local copy.
NOFILL	Do not set terminal fill (same as FILL:0).
QUOTE:yes-no	Do [not] enable control-V character quoting.
RTCOMP:yes-no	Do [not] disable special effects of R and T.
RCVSPEED:n	Set terminal receive speed to n baud.
SBELL:yes-no	Do [not] ring the bell when output is stopped due to exceeding STOP/SSTOP value.

SPEED:dnum	Set receive and transmit speed to dnum baud.
STOP:dnum	Set the terminal STOP counter to dnum.
SSTOP:dnum	Set the terminal SSTOP counter to dnum.
TABS:yes-no	Terminal does [not] have hardware tabs.
TAPE:yes-no	Do [not] allow XON to start paper-tape reader.
TYPE:xxx	Set terminal type to xxx.
UNPAUSE:chr	Set the unpause character to chr.
UC:yes-no	Terminal does [not] have uppercase characters only.
WIDTH:dnum	Set carriage width to dnum columns.
XONXOFF:yes-no	Do [not] allow S/Q to pause output.
XMTSPEED:dnum	Set terminal transmit speed to dnum baud.

Switches of the form /*xxxxxx are unique to one character. The asterisk is NOT part of the switch name. The following is a list of possible arguments which may be accepted by some switches and keywords:

args	A list of keywords and optional arguments.
atxt	A possibly quoted string of ASCII characters. You must include quotation marks if the string contains any characters other than alphanumerics or a dash.
chr	A single, possibly quoted character or an octal constant.
cnum	Core argument: decimal number followed by optional K or P.
dnum	Decimal number followed by optional K, M, or G.
filespec	A standard TOPS-10 file designator of the form dev:file.ext[path].
onum	Octal number followed by optional K, M, or G.
pathspec	A standard TOPS-10 path designator of the form: [project#,programmer#,sfd1,sfd2,...].
prefix	A prefix is the last three characters of the "[LGNxxx ...]" or "%LGNxxx ..." messages.
stxt	A possibly quoted string of SIXBIT characters. You must include quotation marks if the string contains any characters other than alphanumerics or a dash.

yes-no	Switch and keyword arguments may either be NO, YES, OFF (no), ON (yes), 0 (no), or 1 (yes). In addition, you can precede the switch or keyword name with NO to negate its action (such as /NOxxxxxx instead of /xxxxxx:NO).

Characteristics

Does not destroy the core image of either job.

Does not require that you be logged in.

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.

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. You are given job 7; terminal number 116 is attached to your job (7).

```
.LOGIN 27,235<RET>

JOB 7 R5743A SYS #40/2 TTY116
PASSWORD:   <RET>
hh:mm    dd-mmm-yyyy    TUE
```

You attach to an existing detached job (35) with a different PPN. This automatically detaches your current job (7). You enter the correct password at the prompt and LOGIN attaches your terminal to job 35.

```
.ATTACH 35<RET>
PASSWORD:<RET>
FROM JOB 7
```

You attach to job 7 again. You do not need to enter a password because job 7 has your original PPN.

```
.ATTACH 7<RET>
FROM JOB 35
```

You attach to job 35 again, but you mis-type the password. LOGIN does not give you a second chance.

```
.ATTACH 35<RET>
PASSWORD:<RET>
?CAN'T ATTACH TO JOB
```

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

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>

   .

CANCEL Command

Function

The CANCEL command deletes the specified request from the specified queue. You can delete the request before it is started, or you can terminate the request after it has been started.

Format

CANCEL request-type request-id/switch

Where: **request-type** is any of the following:

BATCH-REQUEST
CARD-PUNCH-REQUEST
MOUNT-REQUEST
PAPER-TAPE-REQUEST
PLOTTER-REQUEST
PRINTER-REQUEST

The request-type argument can be abbreviated to a unique set of characters.

request-id is a request identification. This can be any one of the following:

- The request-id number. The request-id number is assigned to the request when it is made, and it is displayed with the SHOW QUEUES command.
- The job name of the request.
- A wildcard construction representing the job names of several requests. The standard wildcards (* and ?) are valid, as well as the percent sign (%) to represent a single character. Wildcards are described in Section 1.11. Note, however, that the asterisk (*) wildcard cannot be used to specify mount requests.

There are two switches that can be used to specify the CANCEL command:

Switch	Function
/HELP	Prints information on your terminal about the CANCEL command. When you use this switch, no requests are cancelled.
/JOBNAME:name	Specifies the name of the job. This switch is used to specify the name of the job when the request-id is the same as the name of another job.

Associated Messages

The following message is printed if the request cannot be cancelled:

NO JOBS CANCELED

This may be the result of trying to cancel a request over which you have no authority.

Characteristics

Runs the QUEUE program.

Destroys your core image.

Example

The following example illustrates the use of the CANCEL, MOUNT, and SHOW QUEUES commands.

A mount request is made for the structure GAL0:

```
.MOUNT GAL0:/NOWAIT<RET>
```

The mount queue is displayed with the SHOW QUEUES command:

```
.SHOW QUEUES MOUNT<RET>

MOUNT QUEUE:
VOLUME  STATUS  TYPE   WRITE    REQ#  JOB#            USER
------ -------  ----  -------  ----  ----  ----------------------
BLKA   WAITING  DISK  ENABLE    650    35  MARTIN,C  [30,5621]
GAL0   WAITING  DISK  LOCKED    672    59  MARY MAROTTA  [27,5434]
THERE ARE 2 REQUESTS IN THE QUEUE
```

To cancel the mount request, the user issues the CANCEL command:

```
.CANCEL MOUNT GAL0<RET>
[MOUNT REQUEST FOR GAL0 CANCELED]
```

Again, the mount queue is displayed:

```
.SHOW QUEUES MOUNT<RET>

MOUNT QUEUE:
VOLUME  STATUS  TYPE   WRITE    REQ#  JOB#            USER
------ -------  ----  -------  ----  ----  ----------------------
BLKA   WAITING  DISK  ENABLED   650    35  MARTIN,C  [30,5621]
THERE IS 1 REQUEST IN THE QUEUE

.
```

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.

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>

System message (see NOTE).

?PLEASE TYPE ^C FIRST
```

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

```
.^C

.CONTINUE<RET>
```

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 which are open for output to be closed. 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. dev: cannot refer to a disk device. 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

Leaves your terminal at monitor level.

Restrictions

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

You cannot specify a disk device as an argument.

Examples

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

```
.CLOSE PTR2:<RET>
.
```

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

```
.CLOSE<RET>
.
```

3. Attempt to terminate I/O from DSKB: assigned logical name DEVA.

```
.ASSIGN DSKB:  DEVA<RET>
 DSKB assigned
 .
 .
 .
.CLOSE DEVA<RET>
 ?Not legal for disk-like devices
.
```

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, .B10	BLISS-10 compiler
.CBL	COBOL compiler
.F4 or .FOR	FORTRAN compiler
.FAI	FAIL compiler
.MAC	MACRO assembler
.PAL	PAL10 compiler
.PAS	Pascal compiler
.P11	MACY11 assembler
.SAI	SAIL compiler
.SIM	SIMULA 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, 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 TMPCOR 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.

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 1.9.)

NOTE

Note that a maximum of 150 processor switch characters can be passed to the compiler.

You can use the following switches as temporary or permanent switches. (Refer to Section 1.8.4.) 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 BLISS-10. 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 source file even if the relocatable binary file is newer. Use this switch to obtain an extra compilation (for example, to obtain a listing of the compilation). NOCOMPILE is the default switch.
/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.
/DEBUG:(arg, arg,...)	Passes the arguments to FORTRAN. Refer to the TOPS-10/TOPS-20 FORTRAN Language Manual.

/DLIST — Creates a listing 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.

/F66 — Applies FORTRAN-66 rules for DO loops and EXTERNAL statements.

/FAIL — Assembles the file with FAIL. Assumed for files with the .FAI extension.

/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 temporary file that contains the library file name and the names of the .REL files produced by the command string. When you issue the FUDGE command, PIP reads this file to generate the library .REL file. (See the TOPS-10 MAKLIB User's Guide for information about library files.)

The argument to /FUDGE is the library file specification. If you omit the file extension, the default is .REL.

This switch is permanent (sticky). That is, it pertains to all .REL files generated by this command string.

/GFLOAT — Indicates that double-precision numbers are to be stored in G-floating format. This format has an expanded exponent range. This option is available on KL10 processors only.

/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 replaced by L or S. To designate a KS processor, use /KS10.

/LIBRARY — Loads the program in library search mode. (/LIBRARY is identical to /SEARCH.)

/LINK — Obsolete

/LIST — Prints the listing file on the line printer (LPT:). If the line printer is spooled or available to your job, the listing file is automatically printed.

/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 .P11. MACY11 is recognized as a compiler 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	Compiles the source file if there is no relocatable file newer than the source file. Note that /COMPILE always compiles, /REL never compiles, and /NOCOMPILE (the default switch) compiles conditionally.
/NODEBUG	Does not pass previously specified arguments to FORTRAN.
/NOLIST	Does not generate listing files. This is the default action. Complement to /LIST.
/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.
/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:.
/PAL10	Assembles the file with PAL10. Assumed for files with the .PAL extension.
/PASCAL	Compiles the file with Pascal. Assumed for files with the .PAS extension.
/REL	Loads an existing .REL file rather than compiling a new one. Refer also to /COMPILE and /NOCOMPILE.
/SAIL	Compiles the file with SAIL. Assumed for files with the .SAI extension.
/SAVE	Saves the core image to disk in an executable file after it is loaded.
/SEARCH	Loads the program in library search mode. (/SEARCH is identical to /LIBRARY.)

/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.
/SIMULA	Compiles the file with SIMULA. Assumed for files with the .SIM extension.
/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	Compiles the program with the compiler from SYS:. This is the default.

Characteristics

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.F10/NEW,SUBRTN.F10/SYS
```

However, the following is valid:

```
.COMPILE TESPRG.F10/NEW
.COMPILE SUBRTN.F10/SYS
```

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.
```

.

4. Compile the files contained in the command file LIBALL. Create a temporary file which contains all the .REL file names and the library file name (MONITR.REL). The FUDGE command creates the library file, MONITR.REL, with the .REL files in the same order as they were specified in the command file.

```
.COMPILE/FUDGE:MONITR.REL@LIBALL<RET>

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

.FUDGE<RET>
```

5. Compile and execute the program NUMBER.FOR.

```
.COMPILE NUMB2.FOR<RET>

FORTRAN:  NUMB2
MAIN.

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

CONTEXT Command

Function

The CONTEXT command displays the status of a job's context. It also allows you to create, kill, and switch between contexts. When issued without arguments, the command gives you information about your current contexts, such as the name of the program in core for that particular context. You can also use CONTEXT to create parallel contexts. Parallel contexts permit you to switch between programs (in separate contexts) without waiting for them to reinitialize. You can specify switches to list the status of a particular context or delete a context. (Section 1.5 contains a discussion of contexts.)

Format

CONTEXT argument/switch

Where: **argument** is optional. With no arguments, CONTEXT displays information about all of your job's contexts, including the program loaded in each context.

/switch is one of the options listed below.

Argument, when specified, can be one of the following:

handle	A context name or number. This argument switches your job's current context to the specified one. A period (.) can replace a context name for the current context only.
name=number	Where name is a one to six character alphanumeric string to be associated with the specified context number.
=	An equal sign can be used to create a new, parallel context, without switching the current context to the newly created context.

/Switch, when specified with the handle argument, can be one of the following:

/KILL	Deletes the specified parallel context.
/LIST	Lists information about the specified context only.

Characteristics

Leaves your terminal at monitor level.

Requires LOGIN.

Preserves your core image.

Restrictions

To use the CONTEXT command, you must be at monitor level, and the job must be halted.

Associated Commands

POP Returns you to a previous superior context, and destroys the current context.

PUSH Creates an inferior context.

Examples

1. Name context 1 (the current context, in this case) TOPLVL.

```
.CONTEXT TOPLVL=1
```

2. Display the status of the current context.

```
.CONTEXT
Contexts used/quota = 1/4, pages used/quota = 0/1000
Contexts     Superior     Prog     Idle time
* TOPLVL 1                PATH
```

Notice that the current context, TOPLVL, is marked by an asterisk (*). For this particular example, one of four allotted contexts is being used, none of the 1,000 saved-pages is in use, the context name is TOPLVL, the context number is 1, and the PATH program is running in this context.

3. Create an adjacent context, run MAIL under it, and exit. Then look at the context status.

```
.CONTEXT=

.MAIL
21 messages, 116 blocks.

MS>quit

.CONTEXT
Contexts used/quota = 2/4, pages used/quota = 4/1000
Context     Superior     Prog     Idle time
TOPLVL  1                PATH      19.78
*       2                MS
```

Note the differences from the previous status. There are now two contexts in use, and the current, unnamed context 2 is running MS. Also, PATH has been idle for 19.78 seconds since the last CONTEXT command was issued. MS will restart very quickly if you type CONTINUE, because it does not have to re-initialize.

4. Finally, kill one context that is running MS, and examine your status. You have to move to another context to do this, because you cannot kill the current context.

```
.CONTEXT 1

.CONTEXT 2/KILL

.CONTEXT
Contexts used/quota = 1/4, pages used/quota = 0/1000
Context     Superior     Prog      Idle time
* TOPLVL   1            PATH
```

Context 1 is now the current and only context.

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

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

        .

        .

        .
```

Later, attach to your original job.

```
.ATTACH 19 [27,5434]<RET>

PASSWORD:<RET>

Type CONTINUE to enter user level.

.CONTINUE<RET>

?PLEASE TYPE ^C FIRST

System message (see NOTE).

Type CTRL/C and CONTINUE to enter user level.

.^C

.CONTINUE<RET>

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

You must 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 1.9.4.

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

Characteristics

Leaves your terminal at monitor level.

Destroys your core image.

Runs the PIP program.

Example

Find file NOTICE.TXT in SYS:

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

Search your own directory.

```
.DIRECT DSKB:NOTICE<RET>
%WLDNSF No such files as NOTICE.TXT
```

Copy NOTICE from SYS: into your directory on DSKB:

```
.COPY DSKB:NOTICE.TXT=SYS:NOTICE.TXT<RET>
```

Check your directory to find NOTICE on DSKB:

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

CORE Command

Function

The CORE command prints or changes the amount of core memory 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. You may not assign more than one section of core to your job.

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.

If you omit nx, the monitor prints the amount of core currently being used by your job, as well as the octal page number, page (accessibility) status, and the origin of the pages. This form of the CORE command does not change the core assignment.

Page status can be any of the following:

- executable (EX)
- readable (RD)
- writable (WR)
- sharable (SH)
- locked (LK)
- allocated-but-zero (AZ).

Origin can be private pages, spy pages, or the file specification.

The total pages in the space is also displayed.

Characteristics

Leaves your terminal at monitor level.

Does not operate when your job is in run state.

Example

Use CORE to look at the amount and contents of memory assigned after using MAIL.

```
.CORE<RET>

Page number     Page status     Origin
0-74            EX RD WR        Private pages
76-165          EX RD WR        Private pages
620-674         EX RD SH        DSKA:MS[1,4]
Total of 162 pages

Virt. mem. assigned 118+45P (Current limit: 16384P Max limit: 16384P)
Phys. mem. assigned 118+45P (Guideline: 16384P Max limit: 999P)
Swap space left: 81502P
```

CPUNCH Command

Function

The CPUNCH command places entries in the card punch queue. Refer to the QUEUE command for further information and examples.

Format

CPUNCH dev:jobname=file-spec/switches

Where: **dev:** is the name of the individual device on which the file is to be punched. (For example, CDP2: is card punch number 2.) The device name is optional. To punch the file on a card punch at a different node, use devSxx;, where xx is the node number. (For example, CDPS22: is a card punch on node number 22.)

jobname is the name of the job you are entering into the queue. The default job name is the name of the first file in the request.

The equal sign is required if you specify either the device or job name.

file-spec is a single file specification or a string of file specifications, separated by commas, for the files being processed. A file specification is in the form dev:file.ext[directory]. (Refer to Section 1.9.)

If you specify neither a job name nor a file-spec, a list of all the jobs in the card punch queue will be printed on your terminal.

The wildcard construction can be used for the file specifications.

/switches are listed below. The switches to this command can be divided into two categories, depending on whether the switch can be used only once, or can be used more times, in a single command string. The two categories are:

- **Queue-Operation Switches**

  These switches can be used only once in a command string. They affect the entire request, and you can place them anywhere in the command string. If you have used one of these switches in a command string, you cannot use it again in the same string. Many commands have a /NO construction, which takes a negative effect. Be sure you do not use the /NO construction of a switch in the same command string with the positive construction.

- **File-Control Switches**

  These switches can be used any number of times in the command string. You can also use the /NO construction of a switch in the same command string. To achieve a temporary or permanent effect by the placement of the switch, refer to Section 1.8.4.

Switches	Category	Function
/ABEFORE: date-time	File control	Queues the file only if the access date is before the specified date and time.
/ACCOUNT:name	Queue operation	Specifies the account to which the job should be charged.
/AFTER: date-time	Queue operation	Processes the request after the specified time.
/ALLFILES: YES or NO	Queue operation	Accepts the request only if all of the files in the request exist. By default, if any of the files do not exist, the others will be processed appropriately. This switch specifies that if any file does not exist, no files should be processed. The value YES or NO is optional. If you use YES, all of the files you specified must exist. If NO, existing files are processed, and warning messages are printed for files that do not exist.
/ASINCE: date-time	File control	Queues the file only if the access date is later than the specified date and time.
/BEFORE: date-time	File control	Queues only the files with creation dates before the specified date and time.
/CHARACTERISTIC: arg	Queue operation	Specifies an output characteristic. You can find a list of the characteristics arguments defined for your system in the file SYS:CHARTY.DAT.
/CHECK	Queue operation	Prints on your terminal a list of the queue entries made by your job.
/COPIES:n	File control	Repeats the output the specified number of times (n must be less than 64). The default is one copy.
/CREATE	Queue operation	Makes a new entry in the specified queue. This function is the default, except when listing queue entries.

/DEFERRED	Queue operation	Causes deferred output to be released to the card punch queue. You must use one of the following switches with /DEFERRED. /CREATE to complete the released output requests. /KILL to eliminate the released output requests. Refer to the SET DEFER command for more information.
/DELETE	File control	Deletes the file after processing it. Same as /DISPOSE:DELETE.
/DESTINATION: node	Queue operation	Specifies the node at which the file will be punched. Use the node name or node number to specify the node. The files will not be punched at any host other than the host to which the terminal is connected.
/DISPOSE:arg	File control	Controls the disposition of the file after it is processed. The arguments to this switch are: DELETE deletes the file from your directory after processing it. PRESERVE preserves the file after processing it. RENAME renames the file from your directory into the spooling area. Thus, the file is effectively deleted immediately.
/DISTRIBUTION: "text"	Queue operation	Specifies text to place in the distribution field, on the banner page of output. You can use this field to include mailing information, or the location where the operator should leave the listing. The text field may be up to 39 alphanumeric characters, including punctuation and spaces if the text is placed in quotation marks.
/ERBINARY	File control	Prints an error message if a binary file is included in the queue. This is the default.
/ERNONE	Queue operation	Prints an error message if no files match the file specification. This is the default.
/ERPROTECTION	Queue operation	Prints an error message if the the request involves a protection violation. This is the default.

/FAST	Queue operation	Prints the entries in the queue on your terminal.
/FILE:arg	File control	Specifies how the file format is to interpreted. The following arguments can be used with this switch: ASCII interprets the file as ASCII text. ELEVEN interprets the file as four 8-bit bytes in each 36-bit word. The bits are arranged as follows: Byte 1: bits 10-17 Byte 2: bits 2-9 Byte 3: bits 28-35 Byte 4: bits 20-27
/FORMS:arg	Queue operation	Specifies any special cards to be used. Available forms are listed in SYS:FORMST.DAT.
/GENERIC	File control	Sends output to the next available card punch. Complement to /UNIT.
/HEADER: YES or NO	File control	Makes header cards for the file. The default is YES.
/HELP:arg	Queue operation	Prints information on your terminal about the QUEUE command. This switch does not queue any files. The switch can be used alone (/HELP) or with one of the following arguments: TEXT prints a message with the format and switches of the QUEUE command. This is the same as /HELP with no arguments. SWITCHES prints a list of all the switches available with the QUEUE command.
/JOBNAME:name	Queue operation	Specifies the name of the job. The name can be up to 6 alphanumeric characters.
/KILL	Queue operation	Removes the specified entry from the queue. You must specify the job name, /REQUESTID, or /SEQUENCE, left of the equal sign in the command line.
/LENGTH:n:m	File control	Processes only the files whose length is between n and m blocks in length.
/LIMIT:n	Queue operation	Limits the output to the specified number of cards.

/LIST:arg	Queue operation	Prints information about the jobs in the queue. If you use /LIST alone, it shows the jobs in the queue. This is equivalent to using the CPUNCH command with no arguments and no switches. /LIST can be abbreviated to /L. The switch can also take one of the following arguments: ALL shows all data about each queue request. FAST shows a fast list of the queue requests. JOBS shows a list of the jobs in the queue. (Same as /LIST with no arguments.) SUMMARY shows only the summary line of the queue display.
/MESSAGE:arg	Queue operation	Specifies the information to be output if an error occurs. The arguments are: ADDRESS — Prints the location in memory where the error occurred. CONTINUATION — Prints information about the error. FIRST — Prints a one-line error message. PREFIX — Prints the six character prefix.
/MODIFY	Queue operation	Alters the specified parameter in the specified job. This switch requires that you have access rights to the job. You must specify the job name, /REQUESTID, or /SEQUENCE, left of the equal sign in the command line. This switch can be used to modify a previously submitted request as long as the request has not been started.
/NEW: YES or NO	File control	Accepts file specifications of files that do not exist.
/NOHEADER	File control	Does not make header cards for each file.
/NONEW	File control	Does not accept file specifications of files that do not yet exist. This is the default.

/NONOTIFY	Queue operation	Does not notify you when the job is finished. This is the default.
/NONULL	Queue operation	Prints a fatal error message on a null request. This is the default.
/NOOPTION	Queue operation	Suppresses reading the SWITCH.INI file.
/NOPHYSICAL	File control	Allows logical names for devices in the command string.
/NOSTRS	File control	When scanning structures for the file, takes only the first occurrence. This is the default function.
/NOTES:"text"	Queue operation	Prints the text in the header card.
/NOTIFY: YES or NO	Queue operation	Notifies you on your terminal when your request is completed. To be notified, use /NOTIFY with no argument, or with YES or 1 as an argument. To suppress notification, use /NOTIFY:0 or /NOTIFY:NO. By default, you are not notified when a request is finished. Special cases, such as printing of batch log files and output of deferred requests, will never notify you when they are completed.
/NULL: YES or NO	Queue operation	Does not print a fatal error message if the specified files do not exist.
/OKBINARY	File control	Accepts files whose extensions indicate that they include binary information. Normally, files with extensions .SAV, .SHR, .LOW, .REL, .EXE, and .HGH will not be accepted for processing.
/OKNONE	Queue operation	Does not produce a warning message if no files match the file specification.
/OKPROTECTION	Queue operation	Does not print an error message if a file protection violation occurs.
/OPTION:name	Queue operation	Uses the option line QUEUE:name in the SWITCH.INI file. SWITCH.INI files are described in Appendix B.
/PHYSICAL	File control	Ignores logical device names in the command line.
/PRESERVE	File control	Saves the file after processing it. This is the default. This switch is the same as /DISPOSE:PRESERVE.

/PRIORITY:n	Queue operation	Assigns the specified priority (n is 1 to 63) to the request. A larger number has greater priority.
/PROTECTION:nnn	Queue operation	Specifies a protection code to be associated with the request. Queue requests may have protection codes. These are exactly like file protection codes. Refer to Section 1.9.4.
/PUNCH:arg	File control	Punches the file in the specified mode. If you omit this switch, the file is punched according to the data mode specified in the file. The following arguments can be used with this switch: 026 — Punches the files in 026 Hollerith mode. ASCII — Punches the files in ASCII card mode. BCD — Punches the files in 026 Hollerith mode. (Same as 026.) BINARY — Punches the files in a checksummed binary card mode. IMAGE — Punches the files in image card format.
/REMOTE	Queue operation	Prints on your terminal a list of remote queues. Must be used with /DESTINATION.
/REQUESTID:n	Queue operation	Specifies the request identification number of the job you wish to modify or terminate. The request identification number is assigned when you queue the request.
/RUN:file	Queue operation	Executes the specified file after the request is accepted.
/RUNCORE:n	Queue operation	Executes the specified file in nK of core after the request is accepted.
/RUNOFFSET:n	Queue operation	Executes the specified file with start offset n after the request is accepted.
/SEQUENCE:n	Queue operation	Specifies a sequence number to aid in identifying a request to be modified or deleted.
/SINCE: date-time	File control	Queues only the files with creation dates after the specified date and time.

/STRS: YES or NO	Queue operation	Searches for the file on all structures in the search list and takes every occurrence. The default is to take just the first occurrence of the file.
/TMPFIL:file: text	Queue operation	Creates a temporary file on TMP: and enters the text into the file.
/UNIT:n	Queue operation	Specifies the unit number of the device you want the output sent to.
/USERNAME: "name"	Queue operation	Specifies the user name field for the banner page of output. This field can contain up to 39 alphanumeric characters, and may include punctuation and spaces if the name is placed in quotation marks.

Associated Messages

When a new entry is made in a system queue, the system prints a message on the user's terminal. The message is in the form:

[CARD-PUNCH JOB name QUEUED, REQUEST #nnn, LIMIT xxx]

Where: **name** is the name of the job in the queue. This can be specified by the user. Otherwise, it defaults to the name of the first file in the request.

nnn is the number that represents the request identification of the job in the queue.

xxx is the maximum number of cards that the job will use.

Characteristics

Leaves your terminal at monitor level.

Runs the QUEUE program.

Destroys your core image.

Does not require LOGIN if you desire a list of queue entries.

Examples

1. Punch the file SYSTAT.MAC in ASCII format.

```
.CPUNCH SYSTAT.MAC/PUNCH:ASCII<RET>
[CARD-PUNCH JOB SYSTAT QUEUED, REQUEST #75, LIMIT 33]
.
```

2. Punch the file SYSTAT.REL in binary format, but do not begin output until after 5:00 P.M.

```
.CPUNCH SYSTAT.REL/PUNCH:BINARY/AFTER:17:00<RET>
[CARD-PUNCH JOB SYSTAT QUEUED, REQUEST #43, LIMIT 200]
```

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.)

Format

CREF file-spec

Where: **file-spec** is a valid file specification. (Refer to Section 1.9.) 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 (*).

You can use the following switches with the CREF command.

Switch	Function
/A	Advances magtape by one file (may be repeated).
/B	Backspaces magtape by one file (may be repeated).
/C	Cancels SWITCH.INI switch defaulting.
/D	Permits default switches as for SWITCH.INI.
/H	Types this text.
/K	Kills user-defined symbol table listing.
/M	Suppresses user macro's, OPDEF's, symbol table.
/O	Lists the opcodes.
/P	Preserves (does not delete) input files.
/R	Restarts listing and prompts for line number.
/S	Suppresses program listing and lists only symbol tables.
/W	Rewinds tape.
/Z	Indicates zero DECtape directory.

Characteristics

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.

```
.SET SPOOL LPT<RET>
.CREF<RET>
CREF:INPUT1
CREF:INPUT2
.
```

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>
CREF:  HIGH
CREF:  SHARE
.
```

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

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.

Example

This is a program to find 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/C's to halt the program.

Type CSTART.

```
.CSTART<RET>
```

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<RET>

FROM JOB 19

        .

        .

        .
```

Later, attach to your original job.

```
.ATTACH 19 [27,5434]<RET>

PASSWORD:<RET>
```

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

```
.CONTINUE<RET>

?PLEASE TYPE ^C FIRST
```

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

```
.^C

.CONTINUE<RET>

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.

DAYTIME Command

Function

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

wkdy dd-mmm-yy hh:mm:ss

Where: **wkdy** is the name of day of the week
dd is the day of the month
mmm is the name of the month
yy is the year
hh is the hour
mm is the minute
ss is the second to the nearest hundredth

Format

DAYTIME

Characteristics

Leaves your terminal at monitor level.

Does not require LOGIN.

Does not destroy your core image.

Example

```
.DAYTIME<RET>
MONDAY 29-FEB-88 16:46:42

.
```

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 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 TOPS-10 DDT Manual.)

Format

DDT

Characteristics

Places your terminal at user level.

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

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<RET>
   TITLE   TEST    SIMPLE PROGRAM
   SEARCH  MACTEN,UUOSYM

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<RET>
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<RET>
EDIT: TEST.MAC

*SJCLF<ESC>JFCL<ESC>.<ESC><RET>
600 TEST: JFCL
*ES<RET>

[DSKC:TEST.MAC]
```

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

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

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 UUO 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!
```

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,UUOSYM
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.
```

DEALLOCATE Command

Function

The DEALLOCATE command removes a volume set from your job's list of allocated resources. DEALLOCATE implies a DISMOUNT of the specified resource. The logical name for the volume set is cleared, and you will not be able to use that logical-name to refer to the volume set.

Format

DEALLOCATE resource-name,resource-name...

Where: For disk volume sets, the **resource-name** is the volume set-name or the structure name.

For tape volume sets, the **resource-name** is the logical name.

DEALLOCATE has one switch:

/HELP Prints a short description of the command. If you specify an argument with this switch, the argument will be ignored.

Associated Commands

ALLOCATE	Informs the system of your future need for a resource.
MOUNT	Requests ownership of a resource.
DISMOUNT	Removes the resource from your job's search list. Dismounts the volume set from the unit if no other users are accessing the resource.
SHOW ALLOCATION	Prints a list of the resources that are allocated and mounted for your job.
SHOW QUEUE	Prints a list of the jobs in the system queues.

Characteristics

Runs the QUEUE program.

Destroys your core image.

Requires LOGIN.

Example

The following example shows the use of the ALLOCATE, DEALLOCATE, MOUNT, DISMOUNT, and SHOW ALLOCATION commands. The resources are reserved for a multivolume tape volume set with the ALLOCATE command. The name of the volume set is TAPE-SET, and it contains three volumes. The logical name TS is assigned to the tape set. The tape is write enabled, and it does not have standard labels.

```
.ALLOCATE TAPE-SET(VOL1,VOL2,VOL3):TS/WRITE-ENABLE/LABEL:NONE<RET>
[ALLOCATE REQUEST TS QUEUED, REQUESTS #672]
```

A file structure named DSKR: is mounted for the job:

```
.MOUNT DSKR:<RET>
[MOUNT REQUEST DSKR QUEUED, REQUEST #673]
[STRUCTURE DSKR MOUNTED]
```

The job's resources are shown using the SHOW ALLOCATION command:

```
.SHOW ALLOCATION<RET>
ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE       ALL  OWN
---------------------- ----------------- ------------ ---  ---
---                    9 TK 800/1600     MAGTAPE UNIT  1    0
---                    RP06              DISK UNIT     2    2
---                    RP20              DISK UNIT     1    1
DSKB                   DSKB              STRUCTURE     1    1
DSKC                   DSKC              STRUCTURE     1    1
DSKR                   DSKR              STRUCTURE     1    1
TAPE-SET               VOL1              MAGTAPE VOL.  1    0
TAPE-SET               VOL2              MAGTAPE VOL.  1    0
TAPE-SET               VOL3              MAGTAPE VOL.  1    0
```

The tape set is mounted, and the resources are again displayed:

```
.MOUNT TS<RET>
[MOUNT REQUEST TS QUEUED, REQUEST #673]
[MAGTAPE TS MOUNTED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE       ALL  OWN
---------------------- ----------------- ------------ ---  ---
---                    9TK 800/1600      MAGTAPE UNIT  1    1
---                    RP06              DISK UNIT     2    2
---                    RP20              DISK UNIT     1    1
DSKB                   DSKB              STRUCTURE     1    1
DSKC                   DSKC              STRUCTURE     1    1
DSKR                   DSKR              STRUCTURE     1    1
TAPE-SET               VOL1              MAGTAPE VOL.  1    1
TAPE-SET               VOL2              MAGTAPE VOL.  1    0
TAPE-SET               VOL3              MAGTAPE VOL.  1    0
```

After work is finished by accessing the tape set and the structure, the structure is dismounted. Because the structure was not explicitly allocated, it is automatically deallocated.

```
.DISMOUNT  DSKR<RET>
[STRUCTURE DSKR DISMOUNTED]
```

The tape volume set is dismounted:

```
.DISMOUNT TS<RET>
[VOLUME SET TS DISMOUNTED]
```

The job's resources are displayed:

```
.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE       ALL  OWN
---------------------- ----------------- ------------ ---  ---
---                    9 TK 800/1600     MAGTAPE UNIT  1    0
---                    RP06              DISK UNIT     1    1
---                    RP20              DISK UNIT     1    1
DSKB                   DSKB              STRUCTURE     1    1
DSKC                   DSKC              STRUCTURE     1    1
TAPE-SET               VOL1              MAGTAPE VOL.  1    0
TAPE-SET               VOL2              MAGTAPE VOL.  1    0
TAPE-SET               VOL3              MAGTAPE VOL.  1    0
```

At this point, the tape set can again be mounted, or it can be deallocated. The tape set is deallocated:

```
.DEALLOCATE TS<RET>
[VOLUME SET TS HAS BEEN DEALLOCATED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE       ALL  OWN
---------------------- ----------------- ------------ ---  ---
---                    RP06              DISK UNIT     1    1
---                    RP20              DISK UNIT     1    1
DSKB                   DSKB              STRUCTURE     1    1
DSKC                   DSKC              STRUCTURE     1    1
```

.

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.

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

Leaves your terminal at monitor level.

Does not destroy your core image.

Requires LOGIN.

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 B<RET>

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 CDR<RET>
```

Again, show the busy devices.

```
.SYSTAT B<RET>

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 compiles the specified source files, loads the resulting .REL files (if necessary), 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.

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 COMPILE, 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). EXECUTE 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.

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 BLISS-10. Assumed for files with the extension of .B10 and .BLI.
/C68 /C74	Runs the specified COBOL version.
/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 TOPS-10/TOPS-20 FORTRAN Language Manual.
/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
/F66	Applies FORTRAN-66 rules for DO loops and EXTERNAL statements.
/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 1.9.
/GFLOAT	Indicates that double-precision numbers are to be stored in G-floating format. This format has an extended exponent range. This option is available on KL10 processors only.
/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 L or S.

/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.) /LIBRARY is the same as /SEARCH.
/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:.
/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.
/PAL10	Assembles the file with PAL10. Assumed for files with the .PAL extension.
/PASCAL	Compiles the file with Pascal. Assumed for files with the .PAS extension.
/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.
/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

Places your terminal at user level.

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

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<RET>
 TITLE TEST SIMPLE PROGRAM
 SEARCH MACTEN,UUOSYM

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<RET>
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<RET>
EDIT: TEST.MAC

*SJCLF<ESC>JFCL<ESC>.<ESC><RET>
600 TEST: JFCL
*ES<RET>

[DSKC:TEST.MAC]
```

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

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

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 UUO AT USER PC 013303
```

Start the debugger, using the DDT command:

```
.DDT<RET>
DDT

13303/ 0  ^
TEST+3/ MONRT.<LF>

PAT. ./ O 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!
```

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,UUOSYM
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.

DECLARE Command

Function

The DECLARE command defines new monitor commands to run specified programs for your job. You can invoke any program with a user-defined command. When you define a command, the monitor searches your command table and then the monitor command table. Exact definitions are given precedence. An exact definition occurs if you type the complete, exact command name. Inexact definitions occur when a command is abbreviated. If there are conflicts within the exact definitions, your definitions are given precedence.

To use abbreviations when typing a command, you can define "uniqueness" for a command. The monitor will behave as if the command had been given an exact definition. If conflicts arise, the monitor searches its own command table without searching your command table.

Format

DECLARE name/switch=filespec

Where: **name** is a command name of 1 to 6 alphanumeric characters.

filespec is the complete file specification of an executable program. There is no default filespec.

/switch is one of the optional switches listed below.

Switches allowed by this command are:

Switch	Function
/AUTOPUSH	Defines a command to automatically PUSH to a new, temporary context, in which the called program will run. When the program is completed, the original context is restored, and the temporary context is destroyed.

NOTE

- Do not define the PATH or DECLARE programs with the AUTOPUSH switch. This is because the original context is restored and all changes made in the temporary context are destroyed when they finish running.

/CLEAR	Clears all user-defined commands. Do not include a command name with this switch.
/KILL	Removes the definition of a command. Requires a command name.
/LIST	Lists the command names currently defined by your job. Do not include a command name with this switch.

/UNIQUE:n	Defines the number of characters in the command that must be typed to be interpreted as your definition and that cannot be overridden by a monitor command. The variable n can be 1, 2, 3, 4, ALL, NONE, or a list of these values separated by commas and enclosed in parentheses. /NOUNIQUE is the same as /UNIQUE:NONE.

Characteristics

Requires LOGIN.

Leaves your terminal at monitor level.

Destroys your core image.

Example

Use DECLARE to define a command, LOOK, to run the SETSRC program.

```
.DECLARE LOOK=SYS:SETSRC.EXE<RET>
```

Use DECLARE to display your job's command list. Other commands in the following example were defined previously.

```
.DECLARE/LIST<RET>

     EDIT DSR    TAPE LOOK
```

Use the new command. Use CTRL/T to display the current state of your job, including the name of the program that is running. Note that, although CTRL/T is shown here, it does not echo on your terminal.

```
.LOOK<RET>

*<CTRL/T>

Day:  1:02:23 Run:  0.13 Rd:32 Wr:0 SETSRC 4+OP T1 PC:002030
Input wait for TTY52:
<CTRL/Z>
EXIT

.
```

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 Section 1.11.)

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

Leaves your terminal at monitor level.

Runs the PIP program.

Requires LOGIN.

Permanently deletes the file(s).

Destroys your core image.

Examples

1. Delete all files with .MAC extension.

```
.DELETE *.MAC<RET>

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

.
```

2. Delete the file TEST.FOR.

```
.DELETE DSKC:TEST.FOR<RET>

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<RET>

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

.
```

DEPOSIT Command

Function

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

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

Requires LOGIN.

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
.
```

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

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.

```
.DETACH<RET>
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, file name 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, DIRECTORY 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 1.14, 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.

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.
/ANYDEVICE	Searches all devices.
/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 1.8.3.
/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.

/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.

/DSKONLY

Searches all disk devices.

/DTA

Lists the directory in old DECtape form.

/EOTS

Stops at the logical end of tape (two consecutive tape marks) when reading a magnetic tape. This is the default. Complement to /NOEOTS.

/ERLOG

Enables automatic device error logging. Complement to /NOERLOG. /ERLOG is the default.

/FAST

Lists short form of directory: file name, extension, structure name, and directory name. Abbreviated to /F. Complement to /NORMAL and /SLOW.

/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.

/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 /HDSDIR 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 DSK:FNDDAT.FDF or .FPF. The files can later be accessed with the /FIND switch. The complement to /FDNBLD is /NOFDNBLD. /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.
/HDSDIR	Prints the device and directory information of the file as a separate header line, immediately preceding the directory listing for each directory. Complementary to /FLSDIR. The default is /NOHDSDIR, unless /WIDTH and /NOFLSDIR are specified.
/HELP:arg	Prints DIRECTORY help text on your terminal. /HELP can be abbreviated to /H. Valid arguments are: TEXT, KEYWORDS, and SWITCHES. TEXT is the default argument; it prints the entire DIRECT.HLP file. The KEYWORDS argument (K) lists and describes all LOGIN switches which take keyword arguments. SWITCHES (S) briefly lists all DIRECTORY switches without explanations. Switches that have a single-letter abbreviation are prefixed with an asterisk.
/INDIRECT	Creates the output listing file in a format suitable for use as a command file to be input to other programs.
/LENGTH:n:m	Processes only files whose length is between n and m blocks.
/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.

/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.

/NOERLOG — Does not enable automatic device error logging. /ERLOG is the compliment to /NOERLOG, and the default.

/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 /HDSDIR or /WIDTH is specified.

/NOFNDBLD — Does not make a Find File from the output. Complement to /FNDBLD; the default is /NOFNDBLD.

/NOHDSDIR — Does not print the device and directory as a header for each directory listed. Complement to /HDSDIR, the default is /NOHDSDIR, unless /WIDTH is specified.

/NOINDIRECT — Does not format the output 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.

/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.
/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 /REWIND.
/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.
/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 /TITLE.
/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.
/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 1.9.1 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 1.9.4.
/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 /NOREWIND.
/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 1.8.3.
/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. TABs are space-filled to maintain a constant number of characters in any given line. 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 /NOTITLE.
/TMPCOR	Lists the directory in old TMPCOR format.
/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 appear in columns. The default for n is the terminal carriage width. (See the SET TTY WIDTH command.)
/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 /ALLOCATED.

Characteristics

Leaves your terminal at monitor level.

Destroys your core image.

Examples

1. List all files on DSKB:.

```
.DIRECT DSKB:<RET>

FILE    DAT    220 <055> dd-mmm-yy 31(225) DSKB: [27,5055]
SWITCH  INI     10 <057> dd-mmm-yy
PROG    MAC      5 <055> dd-mmm-yy

.
```

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

```
.DIRECT *.MAC<RET>

PROG MAC  5 <057> dd-mmm-yy 31(225) DSKB:[27,5055]
FILE MAC  1 <055> dd-mmm-yy 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> dd-mmm-yy 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> dd-mmm-yy BLKT:[7,2,WIZZER]
NEW     ALG     4 <055> dd-mmm-yy
ALGOL   DOC    20 <055> dd-mmm-yy

.
```

5. Show the switches to the DIRECT command.

```
.DIRECT/HELP:SWITCHES<RET>

DIRECT switches are:

ACCESS, ACCOUN, *ALLOC, AUTHOR, CHECKS, COMPAR, DETAIL,
DIRECT, DTA,    EOTS,   ERLOG,  *FAST,  FILES,  FIND,
FLSDIR, FNDBLD, FNDDAT, HDSDIR, *INDIR, *LIST,  MARKS,
MVOLUM, *NORMA, PRDEVI, PRDIRE, PRVERS, RETRY,  REWIND,
SBRMSG, *SLOW,  SORT,   SUMMAR, TITLES, TMPCOR, UNITS,
*WIDTH, WORDS,  WRITTE

Standard switches are:
ABEFOR, ALLOCA, ANYDEV, APPEND, ASCII,  ASINCE, BEFORE,
BINARY, BLOCKS, BYTESI, BUFFER, CONTIG, DATAMO, DELETE,
DENSIT, DSKONL, ERNONE, ERPROT, ERSUPE, ERUID,  ESTIMA,
EXIT,   FIXED,  FRAMES, *HELP,  IMAGE,  IOMODE, LENGTH,
LIB,    MACY11, MECY11, MESSAG, NEW,    NOOPTI, OKNONE,
OKPROT, OKSUPE, OKUID,  OPTION, PARITY, PBEFOR, PHYSIC,
PRINT,  PROTEC, PSINCE, QUERY,  RECSIZ, RECFOR, RUN,
RUNCOR, RUNOFF, SCERRO, SCWILD, SINCE,  STRS,   SUBMIT,
SYS,    TELL,   TMPFIL, VARIAB, VERSIO

.
```

6. Create an output listing file in a format suitable for input to other programs.

```
.
.DIRECT OUT.FIL/INDIRECT=*.TXT

Total of 10 files
.

.TYPE OUT.FIL
DSKB:MAIL.TXT[10,5763]
DSKB:OPRGD.TXT[10,5763]
DSKB:MCO.TXT[10,5763]
DSKB:RDH.TXT[10,5763]
DSKB:MSRDH.TXT[10,5763]
DSKB:MSDPM.TXT[10,5763]
DSKB:INITIA.TXT[10,5763]
DSKB:STEVS.TXT[10,5763]
DSKB:SPIDER.TXT[10,5763]
DSKB:BARRY2.TXT[10,5763]

.
```

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

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 monitor memory. Privileges are required to examine address 41.

```
FILE:  /M<RET>
41/     JSR 3600
 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
```

Enable your privileges.

```
.ENABLE<RET>
```

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

```
.R FILDDT<RET>
FILE:  /M<RET>
41/     JSR 3600
 ^C

.
```

DISMOUNT Command

Function

The DISMOUNT command relinquishes ownership of a device. It does not affect your allocation requests, if you have used the ALLOCATE command, and you can MOUNT the device again. Once you have dismounted a volume set, the device is available to other users. However, if the resource was not allocated with the ALLOCATE command DISMOUNT performs an implicit DEALLOCATE.

If you used the ALLOCATE command for the resource, the DISMOUNTed resource remains in an allocated state until you issue the DEALLOCATE command. Once you have dismounted a volume set, your programs can no longer reference the resource by the logical name except with the MOUNT and DEALLOCATE commands.

Format

DISMOUNT resource-name/switch,resource-name/switch,...

Where: For disk volume sets, the **resource-name** is the volume set-name or the logical name.

For tape volume sets, the **resource-name** is the logical name.

You can use the following switches in the command string:

Switch	Function
/HELP	Prints a short description of the command. If you specify any resource names with this switch, they will be ignored.
/NONOTIFY	Does not set the system to notify your job when the dismount is complete. This is the default function if you have not specified /NOWAIT.
/NOTIFY	Sets the system to notify your job when the dismount is complete. If you specify /NOWAIT, this is the default function.
/NOWAIT	Allows your job to return to monitor level before the dismount is complete. This implies /NOTIFY.
/REMOVE	Removes the specified file structure from your job's search list. If no other jobs are accessing the structure, and if it is not a system structure, the operator physically dismounts it.
/WAIT	Holds your job until the dismount is complete. This is the default function.

Associated Commands

ALLOCATE	Informs the system of your future need for a resource.
DEALLOCATE	Removes the resource from your job's list of allocated resources. If the resource has not been dismounted, DEALLOCATE dismounts it.
MOUNT	Requests ownership of the resource(s).
SHOW ALLOCATION	Prints a list of the resources that are allocated and mounted for your job.
SHOW QUEUE	Prints a list of the requests in the system queues.

Characteristics

Runs the QUEUE program.

Destroys your core image.

Requires LOGIN.

Examples

1. The following example shows the use of the ALLOCATE, DEALLOCATE, MOUNT, DISMOUNT, and SHOW ALLOCATION commands. The resources are reserved for a multivolume tape volume set with the ALLOCATE command. The name of the volume set is TAPE-SET, and it contains three volumes. The logical name TS is assigned to the tape set. The tape is write enabled, and it does not have standard labels.

```
.ALLOCATE TAPE-SET(VOL1,VOL2,VOL3):TS/WRITE-ENABLE
/LABEL:NONE<RET>
[ALLOCATE REQUEST TS QUEUED, REQUESTS #672]
```

A file structure named DSKR: is mounted for the job:

```
.MOUNT DSKR:<RET>
[MOUNT REQUEST DSKR QUEUED, REQUEST #673]
[STRUCTURE DSKR MOUNTED]
```

The job's resources are displayed using the SHOW ALLOCATION command:

```
.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET          RESOURCES          TYPE       ALL  OWN
-------------------- ----------------- ------------ --- ---
---                  9 TK 800/1600     MAGTAPE UNIT  1    0
---                  RP06              DISK UNIT     2    2
---                  RP20              DISK UNIT     1    1
DSKB                 DSKB              STRUCTURE     1    1
DSKC                 DSKC              STRUCTURE     1    1
DSKR                 DSKR              STRUCTURE     1    1
TAPE-SET             VOL1              MAGTAPE VOL.  1    0
TAPE-SET             VOL2              MAGTAPE VOL.  1    0
TAPE-SET             VOL3              MAGTAPE VOL.  1    0
```

The tape set is mounted, and the resources are again displayed:

```
.MOUNT TS<RET>
[MOUNT REQUEST TS QUEUED, REQUEST #673]
[MAGTAPE TS MOUNTED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET          RESOURCES          TYPE       ALL  OWN
-------------------- ----------------  ------------ ---  ---
---                  9TK 800/1600      MAGTAPE UNIT  1    1
---                  RP06              DISK UNIT     2    2
---                  RP20              DISK UNIT     1    1
DSKB                 DSKB              STRUCTURE     1    1
DSKC                 DSKC              STRUCTURE     1    1
DSKR                 DSKR              STRUCTURE     1    1
TAPE-SET             VOL1              MAGTAPE VOL.  1    1
TAPE-SET             VOL2              MAGTAPE VOL.  1    0
TAPE-SET             VOL3              MAGTAPE VOL.  1    0
```

After work with the tape set and the structure is finished, the structure is dismounted. Because the structure was not explicitly allocated, it is automatically deallocated.

```
.DEALLOCATE DSKR<RET>
[STRUCTURE DSKR DISMOUNTED]
```

The tape volume set is dismounted:

```
.DISMOUNT TS<RET>
[VOLUME SET TS DISMOUNTED]
```

The job's resources are displayed:

```
.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET          RESOURCES          TYPE       ALL  OWN
-------------------- ----------------  ------------ ---  ---
---                  9 TK 800/1600     MAGTAPE UNIT  1    0
---                  RP06              DISK UNIT     1    1
---                  RP20              DISK UNIT     1    1
DSKB                 DSKB              STRUCTURE     1    1
DSKC                 DSKC              STRUCTURE     1    1
TAPE-SET             VOL1              MAGTAPE VOL.  1    0
TAPE-SET             VOL2              MAGTAPE VOL.  1    0
TAPE-SET             VOL3              MAGTAPE VOL.  1    0
```

At this point, the tape set can again be mounted, or it can be deallocated. The tape set is deallocated:

```
.DEALLOCATE TS<RET>
[VOLUME SET TS HAS BEEN DEALLOCATED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET          RESOURCES          TYPE       ALL  OWN
-------------------- ----------------- ------------ ---  ---
---                  RP06              DISK UNIT     1    1
---                  RP20              DISK UNIT     1    1
DSKB                 DSKB              STRUCTURE     1    1
DSKC                 DSKC              STRUCTURE     1    1

.
```

2. 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]
```

3. 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

.
```

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

Requires LOGIN.

Leaves your terminal at monitor level.

Does not destroy your core image.

Example

Print the disk usage values for your job.

```
.DSK<RET>
RD,WT=5,10
RD,WT=474,12
```

Print the total disk usage values for job 50.

```
.DSK 50<RET>

RD,WT=474,12
```

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 ACTDAE.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

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/     JSR 3600
 ^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/     JSR 3600
 ^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 MTxn:

Where: **MTxn:** 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 1.9.1 for a description of device name formats.

Characteristics

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>
.
```

EXAMINE Command

Function

The EXAMINE command displays a core location in your area (high or low segment). The contents of the location are typed out in half-word octal mode. This command is useful when debugging programs.

Format

E addr

Where: **addr** is the address of the location being examined. The address is required the first time the EXAMINE command is used.

If you omit the address, the system will examine the next location. If the previous command was a deposit, the system will examine the location of the deposit.

Characteristics

Requires LOGIN.

Leaves your terminal at monitor level.

Requires core.

Example

Examine location 140.

```
.E 140<RET>
000140/ 000000 000000
```

Examine the next location, 141. You do not need to specify the address for a consecutive location.

```
.E<RET>
000141/ 000000 000000
```

Deposit 1 in location 141. You can omit the address because the DEPOSIT command will use the location of the last EXAMINE command.

```
.D 0 1<RET>
```

Examine the same location, 141. You do not need to specify the address. The EXAMINE command will use the location of the last DEPOSIT command.

```
.E<RET>
000141/ 000000 000001
```

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 1.9.)

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 1.8.4.)

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 BLISS-10. Assumed for files with the extension of .B10 or .BLI. This switch is not supported.

/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 TOPS-10/TOPS-20 FORTRAN Language Manual.

/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

/F66 — Applies FORTRAN-66 rules for DO loops and EXTERNAL statements.

/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).

/FUDGE:
file-spec — Creates a temporary file that contains the names of the .REL files produced by the command string plus the library file name. When you issue the FUDGE command, PIP reads this temporary file to generate a library .REL file. (Refer to the FUDGE command description.) The argument to this switch is:

dev:file.ext[proj,prog]

Where: **dev:** is the device on which to write the file. If you omit the device, the system assumes DSK.

file.ext is the name of the library file. The file name is required. If you omit the extension, the system assumes .REL.

[proj,prog] is the directory in which to place the file. Your directory is assumed if you do not specify a directory.

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

/GFLOAT — Indicates that double-precision numbers are to be stored in G-floating format. This format has an extended exponent range. This option is available on KL-10 proessors only.

/K?10 — Designates the processor where the program will execute once it has been loaded. The variable (?) can be L or S. 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.

/LIBRARY — 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 TOPS-10 LINK Reference Manual.)

/LINK — Obsolete

/LIST — 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.

/LMAP — Produces a loader map during the loading process containing the local symbols.

/LOADER — Obsolete

/MACRO — Assembles the file with MACRO. Assumed for files with extensions of .MAC.

/MACY11 — Assembles the file with MACY11. Assumed for files with extensions of .P11. 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).

/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.

/PAL10 — Assembles the file with PAL10. Assumed for files with the .PAL extension.

/PASCAL	Compiles the file with Pascal. Assumed for files with the .PAS extension.
/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<RET>
.COMPILE PARTB.FOR/OLD<RET>
.EXECUTE/REL PARTA,PARTB
```

Characteristics

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

Type out a test program.

```
.TYPE 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.

FILE Command

Function

The FILE command remotely controls DECtape-to-disk and disk-to-DECtape transfers on operator-handled DECtapes.

NOTE

The FILE command is not part of the standard system. Your site must run OMOUNT explicitly to make the FILE command usable.

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.

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 on your terminal with the TYPE 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.

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[ppn],dev:file.ext[ppn]...

   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 writable 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:).

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
.
```

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 instead of the FINISH 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 1.9.1 for a description of device names.

Characteristics

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
```

Show the devices that are busy.

```
.SYSTAT B<RET>

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<RET>
```

Again, show the busy devices.

```
.SYSTAT B<RET>

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. (See the TOPS-10 MAKLIB User's Guide and the TOPS-10 LINK Programmer's Reference Manual for descriptions of library .REL files.)

The temporary file contains the library name and the list of .REL files which were entered by the previous command's /FUDGE switch. The FUDGE command runs PIP, which reads the list of .REL files and the library file name from the temporary file. PIP then copies the listed .REL files into the library file in the same order that they are listed in the temporary file.

NOTE

Unlike the COMPIL program's default action, the /FUDGE switch combines files in the order that you list them in the command line. The COMPIL program sorts files by compilers. COMPIL sorts mixed FORTRAN and MACRO programs so that all FORTRAN programs are compiled first and MACRO programs second.

After 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). Otherwise, the library information in the temporary file will be superseded by the information generated by the other PIP command.

Format

FUDGE

Characteristics

Leaves your terminal at monitor level.

Runs the PIP program (at CCL offset).

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>
```

A DIRECTORY command shows the files created less than an hour ago, among them: LIBRAR.REL.

```
.DIR/SIN:-1<RET>

PROG      REL  1  <055>  dd-mmm-yy  DSKC:[27,5055]
NUMBER    REL  1  <055>  dd-mmm-yy
NUMB2     REL  1  <055>  dd-mmm-yy
LIBRAR    REL  3  <055>  dd-mmm-yy
```

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/switch

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.

/switch is the following option:

/USE:n Specifies the octal section number (n) in which a core image is loaded. Valid section numbers are 0-37, octal. (Note that the limit is 40 octal sections.)

Characteristics

Destroys your core image.

Leaves your terminal at monitor level.

Example

Load a program.

```
.LOAD PROG.FOR
FORTRAN:  PROG
MAIN.
LINK:   LOADING
EXIT
```

Save the executable format.

```
.SAVE
PROG SAVED
```

Use GET to bring the program into memory.

```
.GET PROG
JOB SETUP
```

START execution of the program.

```
.START

TESTING EXECUTION

END OF EXECUTION
CPU TIME:  0.02 ELAPSED TIME:  0.17
EXIT

.
```

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

Places your terminal at monitor level.

Does not require LOGIN.

Examples

1. Request a list of your directory.

```
.DIR<RET>

PATH1.RNO  3  <055>  dd-mmm-yy  DSKC:[27,5434]
SWITCH.INI 1  <055>  dd-mmm-yy
CALC.^C                                    .

^C

.
```

Type two CTRL/Cs to halt output.

2. Execute a program.

```
.EXECUTE PROG.FOR<RET>
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<RET>
```

Use HALT to interrupt execution. (See NOTE below.)

```
.HALT<RET>
.
```

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 site-specific commands, all user-defined (declared) 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.

Characteristics

Does an autopush, so it preserves your core image.

Runs the HELP program.

Leaves your terminal at monitor level.

Does not require LOGIN.

Example

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 site-specific commands, the names of job-specific 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.

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. You can precede any of the following arguments with NO (except NORUN) to suppress a default setting.

Argument	Function
ATTRIBUTES	Prints the terminal attributes for this terminal.
CHECK:keyword	Sets your terminal type after requesting identification from your DEC terminal. The optional keyword is DEFAULT, meaning CHECK only if the current terminal type is the default type (usually TTY).
HELP	Prints information about INITIA command.
KSYS	Tells you when timesharing will end (if KSYS is set).
NAME	Prints the system name on your terminal. NAME is the default argument.
NORUN	Suppresses the execution of any default program.
NOTICE	Prints the general operator notice.
SETTTY	Sets up your terminal characteristics or attributes. The SETTTY argument is required in order to set terminal characteristics from SWITCH.INI if you are logged in. When you are not logged in, INITIA reads terminal characteristics from SYS:TTY.INI automatically. SETTTY can be specified on the INITIA command line or in the SWITCH.INI file.
STRUCTURES	Prints any operator notices about mounted structures.

TEXT	Prints quick operator notice.
TTY	Prints the terminal characteristics for this terminal.

Terminal Characteristics Arguments

In addition to the above arguments, you can use the following arguments to INITIA SETTTY in your SWITCH.INI file. They are effective for any job that is logged in. Most of the following arguments are equivalent to an argument to a SET TTY command. Many of the arguments can be preceded by NO and a space to disable the defaults. These are included in the description of the function of the arguments. (Refer to Appendix B for details about using SWITCH.INI.)

Each argument, equivalent SET TTY command, and function, is listed below.

Argument	Equivalent Command	Function
SETTTY	None	Must be present for the other INITIA terminal 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 carriage returns at the end of output.
CRLF	TTY CRLF	Controls the automatic carriage return and line feed at the end of the terminal line. NO CRLF suppresses the automatic carriage return and line feed.
DEBREAK	TTY DEBREAK	No longer supported.
DEFER	TTY DEFER	Suppresses echoing to a video terminal until input is requested. NO DEFER allows characters typed to the system to be echoed when the terminal is idle.
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.

ELEMENT:xxx	TTY ELEMENT xxx	No longer supported.
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 allows messages when your terminal is at user level.
IDLEDISCONN n	None	Sets the maximum number of seconds your terminal can be idle before the system disconnects it.
LC	TTY LC	Informs the system that your terminal has lowercase ability. NO LC changes all lowercase characters to uppercase.
LENGTH:xx	TTY LENGTH n	Sets the forms length of your terminal.
LOCALCOPY	TTY LOCALCOPY	Does not require the system to echo characters to the terminal.
LOCATE node	None	Changes the default device list of the job, making devices at other nodes available to your job. LOCATE is only valid in the SWITCH.INI and TTY.INI files.
RCVSPEED:xxx	TTY SPEED xxx nnn	Changes the input speed of the terminal to be xxx. This argument affects the speed at which the monitor receives characters from your terminal.
REMOTE	TTY REMOTE	Sets your terminal for accounts with remote or local access types. REMOTE is not recommended for use by non-privileged users because it can only be reversed by an operator [1,2].
RTCOMP	TTY RTCOMP	Controls the use of CTRL/R and CTRL/T. Refer to Section 1.6 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.
SSIZE:n	TTY [S]STOP n	Sets the page length of your terminal. This controls the number of lines that are output to your terminal before an automatic stop.
SSTOP	TTY SSTOP	Ignores CTRL/Q until output is stopped either automatically by the system, or by CTRL/S from the user.
STOP	TTY STOP	Stops output to the terminal automatically after page length is reached.
SYSDPY	None	Runs the display-oriented SYSTAT program that is appropriate for your set terminal type. SYSDPY is only valid in the SWITCH.INI and TTY.INI files.
TABS	TTY TAB	Informs the system that your terminal has tabular ability. NO TABS instructs the system to simulate tab stops.
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 TOPS-10 Monitor Calls Manual. CTRL/S and CTRL/Q are discussed in Section 1.6.
TERMINET	None	Sets tab spacing for a TERMINET-300.
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.

Terminal Attributes Arguments

The following arguments to INITIA SETTTY set up the attributes of your terminal. Like the preceding terminal characteristics arguments, you can use the following arguments in your SWITCH.INI file.

Unlike the terminal characteristics arguments, most of the following terminal attributes arguments do not have equivalent SET TTY commands. The exceptions are DISPLAY, ISO, EIGHTBIT, and OVERSTRIKE.

To display your current terminal attributes, you must use the command INITIA ATTRIBUTES.

To set terminal attributes, use the following command format:

```
INITIA SETTTY arg arg ...
```

All the following arguments can be preceded with NO and a space to disable the setting.

Argument	Function
ANSLEVEL:n	Indicates the level of conformance to ANSI CRT programming standards.
AVO	Indicates that the terminal has an advanced video option or equivalent capability.
BLOCKMODE	Indicates that the terminal is capable of performing block-mode transfers.
COLOR	Indicates the terminal has a color video display.
DECLEVEL:n	Indicates the level of conformance to DEC CRT programming standards.
DECTCS	Indicates the DEC technical character set.
DISPLAY	Indicates that the terminal has a video display; NO DISPLAY indicates a hard-copy terminal.
DRCS	Indicates dynamically redefinable character sets.
EIGHTBIT	Indicates that the terminal uses an eight-bit ASCII character code.
8BITARCH	Indicates the terminal is capable of using an eight-bit ASCII character code.
GUARDEDAREA	Indicates terminal allows guarded area transfer.
HSCROLL	Indicates the terminal supports horizontal scrolling.

IDCHAR	Indicates character insertion and deletion.
IDLINE	Indicates line insertion and deletion.
INTERROGATION	Indicates that the terminal supports terminal state interrogation.
ISO	Indicates ISO Latin-1 supplemental graphic character set. NO ISO means DEC Multinational Character Set.
KATAKANA	Indicates the Katakana character set.
KEYBOARD	Indicates the terminal has a keyboard.
LOCATOR:device	Indicates whether locator is present. The device can be MOUSE, TABLET or NONE.
NRC	Indicates national replacement character sets.
OVERSTRIKE	Indicates that the terminal allows a three-character sequence to create one composite output character by printing one character, backspacing and then printing another character over the first.
PRINTERPORT	Indicates that the terminal has a printer port option.
REGIS	Indicates that the terminal understands ReGIS graphic commands.
SELECTERASE	Indicates selective erase.
SESSIONS	Indicates the terminal supports multiple sessions.
SIXEL	Indicates that the terminal is capable of displaying SIXEL graphics.
SREGION	Indicates scroll regions.
STATUSLINE	Indicates that the terminal has an extra status line in its video display.
TEKEMULATION	Indicates TEK 4010/4014 terminal emulation.
UDKEYS	Indicates that the terminal has user-definable function keys.
USERWINDOWS	Indicates the terminal supports user windows.
VLENGTH	Indicates variable forms length.
VT52EMULATION	Indicates VT52 terminal emulation.
VWIDTH	Indicates variable forms width.

For more information about these terminal attributes, refer to your terminal manuals.

Characteristics

Runs the INITIA program.

Does not require LOGIN.

Example

Use the INITIA TTY ATTRIBUTES command to show the characteristics and attributes of a generic terminal type:

```
.I TTY ATTRIB<RET>

RL357A DEC10 Development 08:11:23 TTY162 system 1026/1042/1322
Connected to Node KL1026(26) Line # 162
Job 20   User DOTY      [27,10024]

TYPE:TTY          MODEL:TTY         CLASS:LT33        APC:HARDWIRED
ECHO:DEFER        WIDTH:72          LENGTH:0          NOSTOP
FILL:1            NOLC              NOTABS            NOFORM
CRLF              NOGAG             SBELL             NOTAPE
BLANKS            ALTMOD            NORTCOMP          NOREMOTE
XONXOF            NOUNPAUS          NOESCAPE          NOQUOTE
IDLEDI:0          EDITOR            NOSLAVE
ANSLEVEL:0        DECLEVEL:0        NOEIGHTBIT        NO8BITARCH
NODISPLAY         NOOVERSTRIKE      NOCOLOR           NOSTATUSLINE
NOISO             NONRC             NODRCS            NOUDKEYS
NOAVO             NOPRINTERPORT     NOIDCHAR          NOIDLINE
NOREGIS           NOSIXEL           NOTEKEMULATION    NOVT52EMULATION
NOSREGION         NOHSCROLL         NOVLENGTH         NOVWIDTH
NOUSERWINDOWS     NOBLOCKMODE       NOGUARDEDAREA     NOSELECTERASE
NOKATAKANA        NOSESSIONS        NODECTCS          NOINTERROGATION
NOLOCATOR         KEYBOARD
```

Use the SET TTY TYPE command to set your terminal type.

```
.TTY TYPE VT240<RET>
```

Use the INITIA TTY ATTRIBUTES command to show your new terminal characteristics and attributes.

```
.I TTY ATTRIB<RET>

RL357A DEC10 Development 08:11:40 TTY162 system 1026/1042/1322
Connected to Node KL1026(26) Line # 162
Job 20   User DOTY      [27,10024]

TYPE:VT240        MODEL:VT240       CLASS:VT200       APC:HARDWIRED
ECHO:DEFER        WIDTH:80          LENGTH:24         NOSTOP
FILL:0            LC                TABS              NOFORM
CRLF              NOGAG             SBELL             NOTAPE
BLANKS            NOALTMOD          NORTCOMP          NOREMOTE
XONXOF            NOUNPAUS          NOESCAPE          NOQUOTE
IDLEDI:0          EDITOR            NOSLAVE
ANSLEVEL:2        DECLEVEL:2        NOEIGHTBIT        8BITARCH
DISPLAY           NOOVERSTRIKE      NOCOLOR           NOSTATUSLINE
NOISO             NRC               DRCS              UDKEYS
AVO               PRINTERPORT       IDCHAR            IDLINE
REGIS             SIXEL             TEKEMULATION      VT52EMULATION
SREGION           NOHSCROLL         NOVLENGTH         VWIDTH
NOUSERWINDOWS     NOBLOCKMODE       NOGUARDEDAREA     SELECTERASE
NOKATAKANA        NOSESSIONS        NODECTCS          NOINTERROGATION
NOLOCATOR         KEYBOARD
```

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

Leaves 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 or more of the following switches. All switches can be preceded with NO (for example, NOTEMP) to negate their functions.

Switch	Function
/BATCH	Deletes files only when you are over the logged-out quota, then deletes enough files to be below quota. The algorithm for determining which files are deleted first is explained in detail below.
/CLEAR	Clears your terminal's display, if it has one and if the monitor knows the terminal type.
/DISCONNECT	Disconnects a terminal line when you log out. This includes hanging up a dataset (modem), or terminating a LAT connection, for example. This switch may be abbreviated to /D. NODISCONNECT maintains the connection with a terminal line after logging you out.
/HELP:key	Prints information about KJOB on your terminal. /HELP does not perform functions 1 through 4. The keywords for /HELP are SWITCHES and TEXT. TEXT prints the help file and is the default argument. SWITCHES lists the KJOB switches and their meanings.

/MESSAGE:key — Tells the system which messages to print on your terminal. NOMESSAGE supresses output of the standard LOGOUT message but still prints error and warning messages. The keywords for /MESSAGE are:

ADDRESS	Prints out the address where the message occurs.
CONTINUATION	Prints out continuation text, if any.
FIRST	Prints out the first line of the message.
PREFIX	Prints out the six-character message prefix.

/TEMP — Searches your disk area and deletes all files with .TMP extensions. TEMP is a default switch for KJOB. Using the NOTEMP switch will log you out faster.

KJOB/BATCH Algorithm

The KJOB/BATCH algorithm categorizes files in three groups, according to their extensions:

- Expendable files have these extensions:

  TMP, TEM, SFD, BAK, Q??, MAP, CRF, LSD, LSQ, LST, LIS, LPT, PTP, PLT, CDP, Z??, FOO, LOG, BIN, DMP, FIN

- Important files have these extensions:

  RNO, RND, RNH, CMD, KBD, CED, MCR, SNO, FAI, FOR, F4, MAC, ALG, AID, BLI, B10, B11, COB, CBL, BAS, PAL, P11, SRC, IDA, IDX, DAT, DBS, B16, B32, B36, REQ, R16, R36, PAS, INI

- Unimportant files have other extensions.

Files are deleted in the following order until the user is under quota:

1. Recompute disk usage and delete files that are too large.

   The number of blocks used is recalculated by doing a wildcard directory and adding up the total space used. Any files that are larger than the logged-out quota are deleted.

2. Delete unprotected expendable files.

   An individual wildcard directory search is performed for each of the extensions listed in the expendable list. If a match is found and the protection code is less than or equal to 177, the file is deleted.

3. Delete unprotected unimportant files.

   Files with extensions that are NOT in the important list and that have a protection code less than or equal to 177 are deleted.

4. Delete remaining expendable files.

   An individual wildcard directory search is performed for each of the extensions listed in the expendable list. If a match is found, the file is deleted, no matter how it is protected.

5. Delete unprotected important files.

   An individual wildcard directory search is performed for each of the extensions listed in the important list. If a match is found and the protection code is less than or equal to 177, the file is deleted.

   Notice that the extensions toward the end of the important list are considered to be more important than those toward the beginning of the list.

6. Delete all unimportant files.

   Files with extensions that are NOT in the important list are deleted no matter what their protection codes.

7. Delete all files.

Characteristics

Deassigns your terminal.

Stops all I/O devices that are assigned to your job.

Runs the LOGIN program.

Example

```
.K<RET>
JOB 99 User CUSTER,L.  [27,4072]
Logged-off TTY64 at hh:mm:ss on dd-mmm-yy
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

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 one or more files 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

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
*FOR21       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 1.9.)

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 BLISS-10. Assumed for files with the extension of .B10 and .BLI.
/C68 /C74	Runs the specified COBOL compiler.
/COBOL	Compiles the file with COBOL. Assumed for files with the extension of .CBL.

/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 TOPS-10/TOPS-20 FORTRAN Language Manual.

/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

/F66 — Applies FORTRAN-66 rules for DO loops and EXTERNAL statements.

/FORDDT — Loads the FORTRAN debugger with the program.

/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]

Where:

dev: is the device on which to write the file. If you omit the device name, DSK: is assumed.

file.ext is the name of the library file. The file name is required. If you omit the extension, it is assumed to be .REL.

[directory] is 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.

/GFLOAT — Indicates that double-precision numbers are to be stored in G-floating format. This format has an extended exponent range. This option is available on KL10 processors only.

/K?10 — Designates the machine on which the program will execute once it has been loaded. The ? can be replaced by L or S.

/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.

/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.

/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.

/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.
/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.
/PAL10	Assembles the file with PAL10. Assumed for files with the .PAL extension.
/PASCAL	Compiles the file with Pascal. Assumed for files with the .PAS extension.
/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.
/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

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.*
TEST FOR 1 <055> dd-mmm-yy  DSKC:  [27,5055]
```

LOAD the program. The monitor compiles and loads TEST.

```
.LOAD TEST.FOR
FORTRAN:TEST
MAIN.
LINK:  LOADING

EXIT
```

Save the loaded program.

```
.SAVE TEST<RET>
TEST SAVED
```

A directory shows that the executable format has been saved in your directory.

```
.DIR TEST.*<RET>
TEST FOR   1 <055> dd-mmm-yy  DSKC:  [27,5055]
TEST REL   1 <055> dd-mmm-yy
TEST EXE 160 <055> dd-mmm-yy  6(422)
  TOTAL OF 162 BLOCKS IN 3 FILES ON DSKC:  [27,5055]
.
```

LOCATE Command

Function

The LOCATE command changes the default device list of the job, making devices at other ANF-10 nodes available to your job. For example, if you want to use the devices at node CHRIS for your job, you can issue the following command:

```
.LOCATE CHRIS
```

Thereafter, the default I/O devices used will be those on node CHRIS. Note that terminal and disk I/O are not affected by this command.

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 ANF-10 node at which you want I/O to be processed.

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

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.

Examples

1. You 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

You use the LOGIN command to gain access to the system. The LOGIN command runs the LOGIN program. LOGIN accepts:

- your project-programmer number (PPN) or your user name
- your password
- optional account and remark strings

You can obtain these from your system administrator. LOGIN can accept a path specification (directory name) in place of the PPN.

To log in, type the LOGIN command, and press RETURN. LOGIN prompts you with a pound sign (#). Type your project-programmer number or your user name, and press RETURN. You can also type your project-programmer number or your user name immediately after typing LOGIN, followed by a space. After it receives this information, LOGIN prints another prompt (PASSWORD:) on your terminal. Type your password, and press RETURN. The password does not echo on your terminal. You are finished logging in when the monitor prompt (a period) is returned.

Your system may be running account validation software. In this case, LOGIN may not be complete after you type your password. If your job requires an account string, LOGIN prompts you for your account (ACCOUNT:). Type your account string, and then press RETURN, or just press RETURN if the system administrator has established a default account for your job. If your job requires a remark string, LOGIN prompts for it (REMARK:). Here you can type anything that will identify your job. Your remark can be up to 39 characters. If you do not have a remark, press RETURN. The system records spaces for your remark. Example 3 shows the procedure for logging in with account validation software.

NOTE

When the account string is validated, characters are checked according to case. Therefore, you must type your account in the same case (uppercase or lowercase) as it is required. This aspect of the system can be changed by the system administrator.

You can put your account string and remark in your SWITCH.INI file. See Appendix B for further information.

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 1.14 of this manual for more information about path specifications.

If you have detached your terminal from a job, and begin 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 identification/switches

Where: **identification** represents either your project-programmer number (PPN), path specification ([directory]), or your user name.

You can type a path specification in place of the PPN or user name. 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 be included 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 description in the TOPS-10 User Utilities Manual.)

Switch	Function
/ACCOUNT:"string"	Specifies the account string for your job. If the account string contains any nonalphanumeric characters, you must enclose the string in quotation marks. This switch is often included in a SWITCH.INI file.
/ASSIGN: (dev1:log1, dev2:log2...)	Assigns a device to your job and a logical name to the device. **dev:** is the device name **log** is the logical name See Section 1.9.1 for more information about device names. The logical name is optional. This switch can be used more than once to assign more than one device.
/ATTACH:argument	Sets the ATTACH state for the LOGIN dialog and the job being created. /ATTACH:ASK is the default action. By default, if a detached job exists with the specified PPN, and the job was logged in with /ATTACH:ASK, LOGIN asks whether you want to attach to the detached job, instead of creating a new job with the same PPN. The /ATTACH:IGNORE switch suppresses the default action of the /ATTACH:ASK switch. Jobs you create with the IGNORE argument will be ignored by LOGIN when you do additional logins with the same PPN. LOGIN will not ask you if you want to attach to jobs set to the IGNORE argument.

/CORE:nx

Informs the system of the maximum amount of core memory that your job can use. The value for x must be either P or K. P represents pages of memory (each page is equivalent to 512 words); K represents blocks of 1024 words. The value n is any decimal number. Refer to the CORE command for further information.

/DEFAULT: (arg:n, arg:n)

Specifies job characteristics. Multiple keywords can be given to the /DEFAULT switch; in that case, they must be enclosed in parentheses and separated with commas. Arguments and their meanings are as follows:

Argument	**Meaning**
BIGBUF:n	Sets the default BIGBUF buffer size for disk to n blocks. Refer to SET DEFAULT BIGBUF command.
BUFFERS:n	Sets the default number of disk buffers to n. Refer to SET DEFAULT BUFFERS command.
PROTECTION:n	Sets the default file protection for your job to n. Refer to SET DEFAULT PROTECTION command.

/DEFER

Defers queueing of spooled output until you log out.

/DSKFUL:arg

Sets the action to be performed if your job exceeds the disk area allowed to it. The arguments are ERROR and PAUSE. If the argument is ERROR, an error condition is passed to your job. This usually terminates the program. If the argument is PAUSE, the program is suspended, and your job is returned to monitor level. This allows you to request operator assistance and then continue the job, as long as you do not issue any commands that destroy your core image. Refer to the SET DSKFUL command.

/DSKPRI:n

Allows privileged users to set the priority for their job's disk operations (data transfers and head positionings). The value can range from -3 to +3. The default timesharing priority is 0. Refer to the SET DSKPRI command.

/GUIDELINE

Specifies that the numeric value cited in the /PHYSICAL switch is a guideline. This is the default setting for /PHYSICAL.

/HELP:keyword	Prints HELP text on your terminal. Valid keywords are: ARGUMENTS, SWITCHES, and TEXT. The ARGUMENTS keyword prints out a list of valid switches and arguments. SWITCHES displays only a list of switches. TEXT will print the entire HELP text. TEXT is assumed if no keyword is supplied. /HELP may be abbreviated to /H.
/LIB:[ppn]	Sets the library area (LIB:) to the specified PPN.
/LIMIT	Specifies that the numeric value specified 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.
/LOCATE:node	Sets the job location to the specified octal ANF node number. Refer to the LOCATE command.
/MAILCHECK	Checks the file DSK:MAIL.TXT to see if you have mail from the MS mail system. You must have DECMAIL/MS on your system. /MAILCHECK is the default action.
/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	/NEW causes NEW:([1,5]) to be searched before SYS:([1,4]) whenever SYS: is specified or implied. If the files are not on NEW:, SYS: will be searched. (See Section 1.13 for more information about NEW:.)
/NODEFER	Does not defer queued output until logout. Refer to the SET DEFER command. This is the default.
/NOMAILCHECK	Suppresses checking the file DSK:MAIL.TXT to see if you have mail from the MS mail system.
/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. When scanning is set, the system searches for files through the entire directory path. Directory paths are described in Section 1.14. This switch disables scanning. Thus, the system will not search for files past the specified directory area. (Refer to the SETSRC program description in the TOPS-10 User Utilities Manual.) /NOSCAN is the default switch.

/NOSETTTY — Instructs the system not to change any of your terminal's characteristics as specified in your SWITCH.INI file. The default action is that LOGIN reads SWITCH.INI, setting terminal characteristics according to switches specified in the LOGIN line. Refer to Appendix B for more information about SWITCH.INI files.

/NOSFDCREATE — Does not create an SFD that was specified as the directory path.

/NOSTR — Suppresses the printing of SYS:STR.TXT.

/NOSYS — Removes the SYS: structure from your DSKspecification, which is your job's search list. Refer to Section 1.12.

/NOTE:file-spec — Prints the specified file after you log in. The file specification must be included; it may contain wild-cards. This switch is useful for printing a project notice file that is kept in a library area. Refer to /NOTICE.

/NOTICE:arg — Controls printing of SYS:NOTICE.TXT and the argument to the /NOTE switch. The arguments for /NOTICE are:

ALWAYS	Always prints notices.
SOMETIMES	Prints notices you have not yet seen. SOMETIMES is the default argument.
NEVER	Never prints notices.

/NOWATCH — Suppresses the printing of incremental job statistics. Refer to the SET WATCH command.

/PASSWORD — Allows you to change your password during the LOGIN procedure. Type in your current password at the Password: prompt. Then type in your new password when the monitor prompts you, and verify it. Your password will then be changed. This switch is ignored under batch. The system manager may, at times, set /PASSWORD on your account, so that when you log in, you will have to change your password.

/PATH:[dir] — Specifies a default path for disk I/O. Refer to Section 1.14 for further information.

/PHYSICAL:nx — Sets the maximum physical page limit of your job. The value n is any decimal number. The value x is either K (for 1024-word blocks) or P (for 512-word pages). You can use either /LIMIT or /GUIDELINE with the /PHYSICAL switch. The default is /GUIDELINE. Refer to the SET PHYSICAL command.

/QUOTA: (str1,str2,...) ALL DSK	Recomputes the disk quota for the specified structures. If you specify more than one structure, you must separate the structure names with commas, and enclose the list in parentheses. However, if you specify only one structure, the parentheses are not required. If you specify no structures, all the structures in your job's search list are assumed. Instead of (str1,str2,...), you can use ALL or DSK. ALL expands to all structures in the system. DSK expands to all structures in your search list.
/REMARK:"text"	Specifies a remark string for your job. If you include non-alphanumeric characters in the remark, you must enclose the remark in quotation marks.
/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. When /SCAN is specified, scanning is enabled for the directory path. Scanning allows searches to be made through the complete path. If scanning is disabled, only the first directory is searched. (Refer to the SETSRC program.)
/SETTTY	Sets your terminal characteristics as specified in the file SWITCH.INI. This is the default function. Refer to Appendix B for more information.
/SFDCREATE	Automatically creates an SFD on the structure, if the SFD was specified as the directory path, and if the SFD does not already exist. This is the default.
/SFDPROT:nnn	Sets the protection of all created SFDs to nnn.
/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 after the description of the switches. 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.
/VIRTUAL:n	Specifies the current virtual page limit, as represented by n. (For a description of CVPL, see the TOPS-10 Monitor Calls Manual.) 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:ALL or /WATCH (arg,arg...)	Duplicates the SET WATCH command, printing messages automatically according to the argument. 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 tilde (~). 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 each line. The width of this line is set with WIDTH. NOCRLF suppresses the automatic carriage-return/line-feed. The default is CRLF.
DEBREAK NODEBREAK	No longer supported.
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.
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 affect 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.
LENGTH:n	Sets the terminal page length to n lines.
RCVSPEED:nnnn	Sets the speed at which your terminal will receive characters to n baud.
RTCOMP NORTCOMP	Controls the function of CTRL/R and CTRL/T. NORTCOMP makes the control characters function as they are described in Section 1.6. 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.
SBELL NOSBELL	Rings the bell when output is stopped automatically by the system. NOSBELL suppresses the terminal bell when output is stopped automatically by the system.
SPEED:nnnn	Sets the receiving and transmitting speed of your terminal to n baud.
SSTOP:n	Sets the terminal to stop output after n lines, where n is page length, ignoring intermittent <CTRL/Q>s.
STOP:n	Sets the terminal to stop output after n lines, where n is page length.
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 1.6.
TIDY NOTIDY	No longer supported.
TYPE:nnnn	Specifies the type of terminal. This informs the system that your terminal is of the type nnnn. This sets some characteristics automatically. For a complete list of supported terminal types, type: HELP * A portion of the information returned is a list of terminal types supported by the monitor.

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.
XMTSPEED:n	Sets the speed at which your terminal will send characters to n baud.
XONXOF NOXONXOF	Allows you to use CTRL/S and CTRL/Q to control the output to your terminal. NOXONXOF prevents the system from stopping terminal output automatically.

Characteristics

Returns your terminal to monitor level or starts a program if specified in ACTDAE.SYS.

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. To gain access to the system, log in with your user name as follows:

```
.LOGIN MCWILLIAMS
JOB 42 RZ373B KL#1026/1042 TTY363
```

If you do not type your project-programmer number or your user name on the same line as the LOGIN command, LOGIN prompts you for that information with a number sign (#).

```
.LOGIN<RET>
JOB 29 RZ373B KL #1026/1042 TTY220
#10,6073<RET>
```

LOGIN prints your assigned job number (job number 29), followed by monitor name, version number, and terminal 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.

```
[LGNLAS Last access to [10,6073] succeeded on
dd-mmm-yy:hh:mm:ss]
hh:mm dd-mmm-yy MON
.
```

If your entries are correct, the system responds with a message indicating when the last attempt to login to your account was, and whether it was successful time, date, day of the week, the message of the day (if any), and a period, indicating readiness to accept another command.

2. The following example illustrates the process of detaching a job, logging in a second job, detaching the second job and logging in a third job. First, detach the job that is currently running:

```
.DETACH<RET>
FROM JOB 52
```

Then log in again:

```
.LOGIN 27,5434<RET>
JOB 54 RZ125A KL #1022/1046 TTY213
PASSWORD: <RET>

OTHER JOBS DETACHED WITH SAME PPN:
    JOB 52 PIP STOPPED

DO YOU WANT TO ATTACH TO THIS JOB?  [Y] NO<RET>
[LGNJSP OTHER JOBS SAME PPN:52]
[LGNLAS Last access to [27,5434] succeeded on
dd-mmm-yy:hh:mm:ss]
hh:mm dd-mmm-yy        MON
```

When you log in, if you have a detached job (logged in with /ATTACH:ASK), LOGIN prompts you whether you want to log in a new job or attach to the existing job. (To suppress this function, use the /ATTACH:IGNORE switch when you log in.) To log in a new job at this point, type NO and press RETURN. If you type YES or just press RETURN, your terminal will be attached to the existing job.

Detaching a job is useful when you are running a program you expect to run for a long time. If you want to allow such a program to run, while you start another job, use the CCONTINUE command. For example, start a DIRECTORY search of SYS:

```
.DIRECT EXEVER.SYS=SYS:*.EXE/PRVERSION<RET>
^C
^C
.CCONTINUE<RET>
```

The DIRECT program continues to search for files in SYS: with the extension .EXE, and stores them in EXEVER.SYS. Meanwhile, you can log in a third job.

To log in a third job, first detach the second:

```
.DETACH<RET>
FROM JOB 54
```

Log in:

```
.LOGIN 27,5434/ATTACH:IGNORE<RET>
JOB 55 RZ125A KL #1022/1046 TTY213
PASSWORD: <RET>

[LGNJSP OTHER JOBS SAME PPN:52,54]
[LGNLAS Last access to [27,5434] succeeded on
dd-mmm-yy:hh:mm:ss]
hh:mm dd-mmm-yy        MON

.
```

The /ATTACH:IGNORE switch was used here to suppress the question about existing detached jobs. To log in a new job, as here, press RETURN.

When the DIRECT program is finished running (as may be seen using SYSDPY), you may want to attach back to job 54. In this case, detach the current job, or log out:

```
.K<RET>

[LGTOUL OTHER USERS LOGGED-IN UNDER [27,5434], JOBS:52,54]

JOB 55 USER MARY MAROTTA [27,5434]
LOGGED-OFF TTY64 AT hh:mm:ss ON dd-mmm-yy
RUNTIME: 0:05:33, KCS:64, CONNECT TIME: 1:45:20
DISK READS: 1534, WRITES: 105, BLOCKS SAVED: 2513
```

Then log in again:

```
.LOGIN 27,5434
JOB 36 RZ125A KL #1022/1046 TTY213
PASSWORD: <RET>

OTHER JOBS DETACHED WITH SAME PPN:
         JOB 52 PIP STOPPED
         JOB 54 DIRECT RUNNING

TYPE JOB NUMBER TO ATTACH OR CARRIAGE-RETURN TO LOGIN
NEW JOB:54<RET>
.ATTACH 54[27,5434]
[LGNATJ ATTACHING TO JOB 52 IN USER MODE]
         TOTAL OF 344 FILES
.
```

LOGIN prints a message after attaching to the running job. This message informs you that the job is in the process of running. Then DIRECTORY prints a message showing the total number of files found.

MAIL Command

Function

The MAIL command starts up a program to send, receive, and store messages to and from other users. For complete details, refer to the TOPS-10/TOPS-20 DECmail/MS Manual.

Format

MAIL

Characteristics

Runs the MS program.

Requires LOGIN.

Destroys your core image.

Places your terminal at user level.

Example

Start up MS, then exit.

```
.MAIL<RET>

MS>EXIT<RET>

.
```

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 despite this warning, 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.

[directory] 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

Place your terminal at user level.

Destroys your core image.

Example

Create a file named TEXT3.MAI.

```
.MAKE TEXT3.MAI<RET>

*EX<ESC><ESC>
```

Exit from TECO.

Use DIRECTORY to see the file.

```
.DIR TEXT3<RET>

TEXT3  MAI   1  <055>  dd-mmm-yy  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] /switch
```

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.

/switch is the following option:

/USE:n Specifies the octal section number, (n), into which a program is merged. Valid section numbers are 0-37, octal.

Characteristics

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
```

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
```

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/ 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, RETURN, or EXIT from MIC processing.
OPERATOR/ 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/ 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, .REVIVE.DIR<RET> 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/ ON	Changes the default action whenever a particular action or event occurs. ON and WHENEVER are equivalent commands.

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<RET>
*TTY:/Q='A.'B<RET>
.DELETE 'A<RET>

.
;A TYPICAL CALL - CHECK IF DSK:C.MIC IS A COPY OF SYS:C.MIC
.DIRECT/CHECK C.MIC.SYS:<RET>


.
.ERROR ?

.R FILCOM<RET>

*TTY:/Q=C.MIC,SYS:<RET>
NO DIFFERENCES ENCOUNTERED

*C

.DELETE C.MIC<RET>
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  dd-mmm-yyyy
FILE 2) SYS:C.MIC      CREATED: 1202  dd-mmm-yyyy

?FILES ARE DIFFERENT

*[ABORT ON ERROR]
^C

.
```

MOUNT Command

Function

The MOUNT command requests ownership of a device. Because MOUNT calls for action by the operator, the command is not complete (the monitor prompt is not printed on your terminal) until the operator has mounted the device. If more than one volume in the volume set must be mounted at one time, the system ensures that the resources will be available. Volumes that must be mounted sequentially, such as tape volume sets, will be mounted automatically. Therefore, you use MOUNT only once for each volume set.

Resources are pre-allocated to a job using the ALLOCATE command. This informs the system of your future need for a resource. If you use the MOUNT command without previously using the ALLOCATE command for the same resource, MOUNT allocates the resource as well as mounting it. Note, however, that the implicit MOUNT allocation is cancelled when you use DISMOUNT. An explicitly allocated resource remains your property until you use DEALLOCATE to relinquish it.

You can use MOUNT to assign a logical name to a resource. A tape volume set must always have a logical name. Therefore, if you do not assign a logical name when you allocate or mount a tape volume set, the system uses the first six alphanumeric characters, or up to the first non-alphanumeric character, as the default logical name.

Format

```
MOUNT resource:log-name/switch/switch...
```

Where: **resource** is one of the following:

- Disk structure or volume set name, such as DSKB:.
- Tape volume set and identifiers, such as PAY-WK4:(PM34,PM35), where PAY-WK4 is the name of the volume set, and (PM34,PM35) is a list of the names of each tape in the volume set.
- A logical name that was previously associated with a resource.
- A physical device name.

Note that a tape mount request must include a volume identification. This is accomplished by including the volume identifier in the resource name, or by using the /SCRATCH, /VOLID or /REELID switch in the command line.

log-name is the logical name you assign to the device that you will use. The logical name can be up to 6 alphanumeric characters. The logical name is optional. Disk volume sets do not require a logical name, but you can assign one. Tape volume sets must have a logical name. If you do not assign a logical name to a tape volume set, the default logical name will be derived from the volume set name, using the first six characters, or up to the first non-alphanumeric character.

If you assigned a logical name to the resource by using the ALLOCATE command, you can mount the device by typing:

MOUNT log-name/switch/switch...

Any switches that you specified in a previous ALLOCATE command, using the same resource name, become effective when you MOUNT the resource. However, you cannot specify any switches with MOUNT to change the switches you specified in the ALLOCATE command. Also, you cannot specify any further switches with the MOUNT command, except for non-status-setting switches such as /REMARK. This is because defaults are assumed when the request is ALLOCATEd.

NOTE

To MOUNT more than one device at a time, separate the volume set identifications with a comma (,).

You can obtain a list of all the requests in the mount queue by typing MOUNT with no arguments or switches.

The following is a list of the switches you can use with MOUNT. Some switches apply to both tape and disk volume sets, other switches apply to one or the other. The center column indicates the kind of volume set the switch applies to.

Switch	Device	Function
/ACTIVE	Disk	Requests that the volume set be placed in your job's active search list. (Refer to the SETSRC program description in the TOPS-10 User Utilities Manual.) The structure will become part of the list that the system automatically uses to search for a file. This is the default function. This switch is the complement to /PASSIVE.
/CHECK	Tape Disk	Prints on your terminal a list of the mount requests made by your job.
/CREATE	Disk	Allows files to be created on the structure. This switch is the complement /NOCREATE. This switch implies the /ACTIVE switch.
/DENSITY:n-BPI	Tape	Specifies the recording density (bits-per-inch) of the volume set. The argument (n) can be: 200, 556, 800, 1600, or 6250. The -BPI portion of the value is optional.
/DISK	Disk	Identifies the volume set as a disk volume set.
/EXCLUSIVE	Disk	Ensures that you will have exclusive access to the volume set. No other users will be allowed to access the resource. You must have the same project number as the owner of the volume set.

/HELP	Tape Disk	Prints a brief description of the MOUNT command.
/LABEL-TYPE: arg	Tape	Specifies the kind of label processing to be used and indicates the label status of the tape. The label type is used to ensure that the correct tape has been mounted. The arguments and their meanings are:
		ANSI — The label is formatted according to ANSI standards.
		BLP BYPASS — The tape may or may not have labels, but it is treated as if it were unlabeled. This argument can only be used by privileged users.
		EBCDIC IBM — The label is formatted in EBCDIC
		NOLABELS NONE UNLABELED — The tape does not have a standard label. You will not be informed when the end-of-tape is reached.
		USER-EOT — The tape does not have standard labels. However, it may have user labels. When the end-of-tape is reached, the user is notified. This is useful for programs such as BACKUP.
/NEW-VOLUME-SET	Tape	Specifies that a new volume set is going to be created. The operator will choose a new tape or tapes for you. This switch implies /WRITE-ENABLE.
/NOCREATE	Disk	Prevents the creation of files on this volume set unless you specify the volume set when you write the files. This switch is the complement to /CREATE and it implies /ACTIVE.
/NONOTIFY	Tape Disk	Does not notify you when your request is finished. This is the default function.

/NOTIFY	Tape Disk	Sets the system to inform you when the resource is mounted or dismounted. The system sends a message to your terminal when any of the following occurs: o The resource is physically mounted. o The resource is dismounted by the operator without a request by your job. o Another volume in a tape volume set is mounted. o The disk structure is locked or unlocked by the operator.
/NOWAIT	Tape Disk	Allows you to continue working on the system before the resource is mounted. This switch implies /NOTIFY and it is the complement to /WAIT.
/PASSIVE	Disk	Requests that the structure be placed in your job's passive search list. (Refer to the SETSRC program.) The system will not search for files on this structure unless you specify the structure name in the file specification. This switch is the complement to /ACTIVE.
/QUOTA	Disk	Recomputes the usage quota on the specified structure.
/READ-ONLY	Tape Disk	Specifies that you will not write on the volume set. Tape volume sets will be checked as they are mounted, to ensure that they are not write-enabled. This is the default for tape volume sets. On disk volume sets, the monitor will not update access dates. If you specify /EXCLUSIVE and /READ-ONLY, the operator may write-protect the volume set. This switch supersedes /RONLY, /WLOCK, and /WRITE:NO.
/REMARK:"text"	Tape Disk	Allows you to send a message to the operator identifying the volume to be mounted. The argument (text) can be up to 50 characters long. Use quotation marks if the text contains spaces or punctuation.
/SCRATCH	Tape	Instructs the operator to mount a scratch tape. The operator will select a tape that is blank, with the intention of keeping the tape when you are finished with it. This switch implies /WRITE-ENABLE.

/SHARABLE	Disk	Allows other users to access the resource. This is the default function. This switch is the complement to /EXCLUSIVE.
/TAPE	Tape	Specifies that the volume set is a tape volume set. This switch is required when the volume set has the same name as a cataloged disk volume set.
/TRACKS:n	Tape	Specifies the number of tracks on the tape. The value of n can be 7 or 9.
/USER:[ppn]	Tape Disk	Prints on your terminal a list of all requests for a particular user.
/VOLID:volid	Tape	Identifies the volumes in a tape volume set. This switch can be used only if the volid-list was not specified previously. If the volume set is comprised of more than one volume, the volids should be separated by commas, and the volid-list should be enclosed in parentheses. This switch supersedes /REELID.
/WAIT	Tape Disk	Ensures that the volume set will be mounted before you continue working on the system. This is the default function. This switch is the complement to /NOWAIT.
/WRITE-ENABLE	Tape Disk	Ensures that you can write on the volume set. For tape, the monitor checks each volume as it is mounted, to be sure that it is write-enabled. This is the default function for disk volume sets.

Associated Commands

ALLOCATE	Informs the system of your future need for a resource.
DISMOUNT	Removes the specified volume set from your job search list. If no other users are accessing the resource, and it is not a system structure, the volume set will be dismounted from the unit.
DEALLOCATE	Removes the resource from your job's list of allocated resources.
SHOW ALLOCATION	Prints a list of the resources allocated and mounted for your job.
SHOW QUEUE	Prints a list of the jobs in the system queues.

Characteristics

Runs the QUEUE program.

Destroys your core image.

Requires LOGIN.

Example

The following example shows the use of the ALLOCATE, DEALLOCATE, MOUNT, DISMOUNT, and SHOW ALLOCATION commands. The resources are reserved for a multivolume tape volume set with the ALLOCATE command. The name of the volume set is TAPE-SET, and it contains three volumes. The logical name TS is assigned to the tape set. The tape is write enabled, and it does not have standard labels.

```
.ALLOCATE
TAPE-SET(VOL1,VOL2,VOL3):TS/WRITE-ENABLE/LABEL:NONE<RET>
[ALLOCATE REQUEST TS QUEUED, REQUESTS #672]
```

A file structure named DSKR: is mounted for the job:

```
.MOUNT DSKR:<RET>
[MOUNT REQUEST DSKR QUEUED, REQUEST #673]
[STRUCTURE DSKR MOUNTED]
```

The job's resources are shown using the SHOW ALLOCATION command:

```
ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE       ALL  OWN
---------------------- ----------------  ------------ ---  ---
---                    9 TK 800/1600     MAGTAPE UNIT  1    0
---                    RP06              DISK UNIT     2    2
---                    RP20              DISK UNIT     1    1
DSKB                   DSKB              STRUCTURE     1    1
DSKC                   DSKC              STRUCTURE     1    1
DSKR                   DSKR              STRUCTURE     1    1
TAPE-SET               VOL1              MAGTAPE VOL.  1    0
TAPE-SET               VOL2              MAGTAPE VOL.  1    0
TAPE-SET               VOL3              MAGTAPE VOL.  1    0
```

The tape set is mounted, and the resources are again displayed:

```
.MOUNT TS<RET>
[MOUNT REQUEST TS QUEUED, REQUEST #673]
[MAGTAPE TS MOUNTED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE       ALL  OWN
---------------------- ----------------  ------------ ---  ---
---                    9TK 800/1600      MAGTAPE UNIT  1    1
---                    RP06              DISK UNIT     2    2
---                    RP20              DISK UNIT     1    1
DSKB                   DSKB              STRUCTURE     1    1
DSKC                   DSKC              STRUCTURE     1    1
DSKR                   DSKR              STRUCTURE     1    1
TAPE-SET               VOL1              MAGTAPE VOL.  1    1
TAPE-SET               VOL2              MAGTAPE VOL.  1    0
TAPE-SET               VOL3              MAGTAPE VOL.  1    0
```

After work is finished by accessing the tape set and the structure, the structure is dismounted. Because the structure was not explicitly allocated, it is automatically deallocated.

```
.DISMOUNT DSKR<RET>
[STRUCTURE DSKR DISMOUNTED]
```

The tape volume set is dismounted:

```
.DISMOUNT TS<RET>
[VOLUME SET TS DISMOUNTED]
```

The job's resources are displayed:

```
.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE       ALL  OWN
---------------------- ----------------- ------------ ---  ---
---                    9 TK 800/1600     MAGTAPE UNIT  1    0
---                    RP06              DISK UNIT     1    1
---                    RP20              DISK UNIT     1    1
DSKB                   DSKB              STRUCTURE     1    1
DSKC                   DSKC              STRUCTURE     1    1
TAPE-SET               VOL1              MAGTAPE VOL.  1    0
TAPE-SET               VOL2              MAGTAPE VOL.  1    0
TAPE-SET               VOL3              MAGTAPE VOL.  1    0
```

At this point, the tape set can again be mounted, or it can be dismounted and released from your job. The tape set is deallocated:

```
.DEALLOCATED TS<RET>
[VOLUME SET TS HAS BEEN DEALLOCATED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE       ALL  OWN
---------------------- ----------------- ------------ ---  ---
---                    RP06              DISK UNIT     1    1
---                    RP20              DISK UNIT     1    1
DSKB                   DSKB              STRUCTURE     1    1
DSKC                   DSKC              STRUCTURE     1    1

.
```

NETWORK Command

Function

The NETWORK command prints information about the nodes in an ANF-10 and/or DECnet-10 network environment. Your terminal must be attached to one of the nodes in the network to use this command. You can use the NETWORK command to obtain information about the other nodes in either type of network, and about the device configuration of each node in the ANF-10 network.

You can specify the node-name or node-number in the ANF-10 network, or just node-name in the DECnet-10 network, to limit the output to specific nodes. You can use switches to change the output format.

NETWORK reads your SWITCH.INI file and accepts indirect commands. The indirect command files should be formatted as follows:

node-list/switch/switch...

Indirect command files are discussed in Appendix C.

Formats

NETWORK node-list/switch

Where: **node-list** is the node-names or node-numbers (ANF-10), or just node-names (DECnet-10), of nodes for which you want information. If you do not specify the node-list, information is output for every node in the network. The node-names or node-numbers (ANF-10), or just node-names (DECnet-10) are separated by commas.

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.

NOTE

You use switches to select a node with or without a specific attribute. Use the proper switch, or prefix the switch with NO. If you do not use the switch, the selection of a node does not depend upon that attribute. If you do not specify a node, the default output is /BRIEF.

The command line takes the format specified below:

NETWORK node-name,
node-number,.../switch/switch...

You can repeat node names, but not switches.

The switches to the NETWORK command follow.

Output Control Switches

Switch	Function
/ANF10	Forces output of ANF-10 information.
/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.
/COST	Controls the listing of the physical link "cost" associated with the /TOPOLOGY switch.
/DECNET	Forces output of DECnet-10 information.
/ERROR	Prints all error messages. This is the default.
/FAST	Prints only the name and number of the nodes. If you also specify /NOBRIEF, the configuration information is printed.
/HEADER	Prints header information. This is the default action.
/HELP	Prints a description of the NETWORK command.
/LAT	Prints LAT terminal server information.
/NOANF10	Suppresses output of ANF-10 information.
/NOBRIEF	Prints a list of the devices on each node. This is the default if you specify a node-list.
/NOCOST	Suppresses the listing of the physical link "cost" associated with the /TOPOLOGY switch.
/NODECNET	Suppresses output of DECnet-10 information.
/NOERROR	Suppresses the output of error messages.
/NOFAST	Suppresses printing the name and number of the nodes. If you also specify /NOBRIEF, the configuration information is suppressed.
/NOHEADER	Suppresses output of header information.
/NOLAT	Suppresses LAT terminal server information.
/NOSILENCE	Prints all information. This is the default.
/NOSORT	Suppresses sorting of the nodes according to node number.
/NOTOPOLOGY	Suppresses printing of the topology of the network.
/SILENCE	Prints only error messages.
/SORT	Sorts the nodes according to node number.

/TOPOLOGY	Prints the topology of the network as a node name and number followed by its neighbors' numbers. Each neighbor has a physical link "cost" following it in parentheses unless you also include the /NOCOST switch.

ANF-10 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.
/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.

DECnet-10 Node Selection Switches

Switch	Function
/LINKS	Prints only nodes that have one or more open logical links.
/NOLINKS	Prints only nodes that do not have one or more open logical links.
/NOUNREACHABLE	Suppresses listing of all known DECnet nodes.
/UNREACHABLE	Lists all known DECnet nodes.

Network Topology

The topology of a network is determined by the interconnection of nodes in a network. Nodes can be adjacent to each other connected by a physical link. They can also be connected with intermediate nodes. These connections, with associated costs, determine the routes (paths) a message takes from one node to another. The costs are arbitrary values given to each physical connection of the network. The cost of a given route is determined by the sum of the costs of the physical connections along that route. In cases where multiple routes exist between nodes, the route with the lowest cost is taken. In the ANF-10 network, this information is available when you use the /TOPOLOGY switch.

In the DECnet network, the /TOPOLOGY switch displays the information available for DECnet nodes. The first two items of information are the node name and number, followed by DECnet routing information for that node. If the node is unreachable, then this information is placed in the output line: "Unreachable." The remaining information columns are left blank to indicate that information is not available. If the node is reachable, then the column shows the circuit name for the initial routing of a message to that node.

The following columns show cost and number of hops, analogous to the ANF-10 network topology. The cost for DECnet is set through network management functions on a per physical-connection basis. If you ever form a logical link between your node and the destination node, information is available for the count of open logical links and the round-trip delay time in milli-seconds. Otherwise, this information is not available, and the information fields are left blank. Example 4 shows the output from the use of the /TOPOLOGY switch for both ANF-10 and DECnet-10 network environments.

Associated Messages

The output from the NETWORK command is in one of the following formats.

When you do not specify a node-list:

```
For ANF-10:
  node-name (node-number) system-name system-generation-date
For DECnet-10:
  Node list (node-name)
```

When you specify a node-list:

```
For ANF-10:
  node-name (node-number)  system-name system-generation-date
device[number-of-devices] device[number-of-devices]
device[number-of-devices]...
For DECnet-10:
  same as when you don't specify node names.
```

Characteristics

Runs the NETWORK program.

Does not require that you be logged in.

Examples

1. The following example shows the NETWORK command and output when you omit the node-list. There are 17 nodes in the ANF-10 network; five are shown in this example. There are 183 reachable nodes in the DECnet network; 14 are shown in this example.

```
.NETWORK<RET>

[ANF10 network:  connected to SPIRIT (30), located at
KL1026(26), 17 nodes]
Node CTCH22   (22)    DN82 V23(174)          dd-mmm-yy
Node KL1026   (26)    RC117B KL #1026/1042 dd-mmm-yy
Node NEXT     (27)    DN87 V23(173)          dd-mmm-yy
Node SPIRIT   (30)    DN87 V23(174)          dd-mmm-yy
Node NOVA     (31)    DN87S V23(174)         dd-mmm-yy

[DECnet network: local node KL1026, 183 reachable nodes in
area 7]
ABLE     ADAM     AJAX     ALGOL    ALIEN    ALPHA    ALPINE
BLUE     CACHE    CADVAX   CAR      CASTOR   CDR      CHAOS
```

2. The following example shows the NETWORK command when the node-list is specified. Node-number 33 is valid; node 76 is invalid.

```
.NETWORK 33,76<RET>
[ANF10 network: connected to NOVA(31), located at KL1026(26),
15 nodes]
Node     DWARF     (33)    DN87S V23(174)  28-Sep-82
         TSK[2] TTY[64]

%NWKNNN Node 76 not in Network

.
```

3. The following example shows the use of NETWORK with the /NOTSK switch.

```
.NETWORK/NOTSK<RET>

[ANF10 network: connected to NOVA(31), located at KL1026(26),
15 nodes]
Node    SOFDCP  (75)    DN82 V22E/52A
Node    WOBBLE  (143)   DN81 V22E/52A

[DECnet network: local node KL1026, 182 reachable nodes in
area 7]
ABACUS  ABLE    ADAM    AJAX    ALGOL   ALIEN   ALPHA   ALPINE
ANIMAL  ARK     BAXTER  BERGIL  BISON   BLUE    CACHE   CADVAX
CAR     CASTOR  CDR     CHAOS
```

4. The following example shows the use of the NETWORK command with the /TOPOLOGY switch for both ANF-10 and DECnet networks. The connected node (the node running your job) is KL1026. KL1026 has a direct line to node ENCORE(32), node NOVA(31), node JINX(34), node DWARF(33), and node NEXT(27). Each of these lines costs 10. Therefore, they are equally acceptable links. Note that node NEXT(27) has a direct line to node KL1026(26), but this line costs 62, and is therefore less acceptable than the line which connects node KL1026 to node NEXT.

```
.NETWORK/TOPOLOGY<RET>
[ANF10 network: local node KL1026(26), 17 nodes]

Node    KL1026  (26)    32(10)  34(10)  33(10) 27(10)
Node    CTCH22  (22)    76(8)
Node    KS4101  (76)    22(10)  27(10)
Node    SOFDCP  (75)    71(8)   6(8)
Node    TWINKY  (71)    75(10)  123(10)
Node    WIZARD  (123)   71(8)   27(8)
Node    NEXT    (27)    26(62)  123(16) 76(16)
Node    DWARF   (33)    26(62)
Node    ENCORE  (32)    26(62)
Node    JINX    (34)    26(62)
Node    NOVA    (31)    26(62)  20(8)
```

The DECnet network /TOPOLOGY switch displays cost and number of hops between nodes.

```
[Decnet network: local node KL1026, 183 reachable nodes in
area 7]
Name    Number    Line          Cost   Hops L.Links Delay
LARRY   (7.221)   DTE-0-3       7      3
LYRA    (7.236)   ETH-0         1      1

.
```

5. The following example shows the use of the NETWORK command for a DECnet Ethernet endnode. An Ethernet endnode does not know DECnet topology information, so the system prints a summary line containing DECnet information.

```
.NETWORK/DECNET<RET>
[DECnet network:local node KL1026,
running as an Ethernet endnode]
```

NODE Command

Function

The NODE command prints ANF-10 and DECnet 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.

To use some commands (for example, SET HOST), you must know if a node has a command interpreter. The NODE command displays the symbol MCR as a device on each ANF-10 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

Leaves your terminal at monitor level.

Does not destroy your core image.

Associated Messages

When you specify a node-id, the information for ANF-10 nodes is displayed in the following format:

node type 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.

Information for DECnet nodes is displayed in one of the following formats:

- DECnet node-name (node-address) HOPS:m COST:n VIA circuit name
- DECnet node-name (node-address) may be reachable via the area router

- o DECnet node-name (node-address) may be reachable via the designated router

- o DECnet node-name (node-address) unreachable

Where node-name is the name of the DECnet node, node-address is the DECnet area and node number, HOPS and COST refer to the length of the path to that node, circuit is DTE-cpu-lineno or ETH-0 (on a KL) or KDP-0-lineno (on a KS), and cpu and lineno are the numbers of the CPU and line through which the node is reachable. If the node is in a different DECnet area, or the system is running as an Ethernet endnode, you will get either the second or the third message. The node is classified as unreachable only if it is in the same area, but not currently running.

Examples

1. The following example shows the command interpreter to which your terminal is connected.

```
.NODE 0<RET>
Local       KL1026(26)     RZ357A KL  1024/1042 mm-dd-yy
            MCR[1] TTY[137] CDR[2] LPT[5] PTR[2] PTP[2] PLT[1]
.
```

2. Print the information associated with various DECnet and ANF-10 nodes:

```
.NODE KL2116<RET>

DECnet  KL2116(7.116) Hops:6 Cost:11 via DTE-0-3
.
```

The message indicates KL2116 is available, via DTE.

```
.NODE ELROND<RET>

DECnet  ELROND(4.19) May be reachable via the area router
.
```

The message indicates ELROND may or may not be reachable, and is in a different area. You can only tell if ELROND is running by attempting a connection.

```
.NODE NOVA<RET>

ANF     NOVA(31) DN87S V24(226) dd-mmm-yy
TTY[63] LPT[1] TSK[2]
.
```

The message indicates NOVA is an ANF-10 node, and tells you what devices it has.

```
.NODE VLNVAX<RET>

DECnet  VLNVAX(7.132) Hops:1 Cost:1 via ETH-0
.
```

The message indicates VLNVAX is available, via Ethernet.

3. Print information about a node named BLAND. The message indicates the node does not exist.

```
NODE BLAND<RET>

?Undefined Network Node
```

PASSWORD Command

Function

The PASSWORD command allows you to change your current password. Your password may be a maximum of 39 characters. Your system administrator may set a minimum password length.

It asks you to confirm the change by typing your new password again. If the two (new password and verification) do not match, the password is unaltered. Neither old password, new password, nor the verification echoes on the terminal. The command prompts you for all input.

Format

PASSWORD

Characteristics

Leaves your terminal at monitor level when all required information is supplied.

Requires LOGIN.

Destroys your core image.

Example

This is an example of a successful password change.

.PASSWORD<RET>

You will then be prompted as follows: (Remember, what you type is not echoed on the terminal.)

```
Password:  <RET>
New password:  <RET>
Verification:  <RET>

.
```

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

Leaves your terminal at monitor level.

Does not destroy your core image.

Example

Show information about your job.

```
.PJOB<RET>
JOB 42    USER MANJUSREE  [10,6000]   TTY101
.
```

PLEASE Command

Function

The PLEASE command allows you to communicate with the operator. You can send a message to the operator, leaving your job at monitor level, or you can enter a dialogue that allows two-way communication with the operator.

Formats

PLEASE text<ESC>

This format sends a message to the operator and leaves your job at monitor level.

PLEASE<RET>

This format enters into a dialog with the operator. This format is also useful for a message that is more than one line long. Refer to Example 1.

When you enter dialog mode, you can enter your text either of two ways:

- If you type your message and end it with <CTRL/Z> (control-Z), your job will wait for a reply from the operator.
- If you type your message and end it by pressing ESCape, the message will go to the operator and your job will be returned to monitor level. You may press <ESC> without typing a message. Your job will exit to monitor level immediately.

The PLEASE command has two switches:

Switch	Function
/HELP	Prints information on your terminal about the PLEASE program.
/NODE:id::	Specifies the node of the operator you wish to communicate with. Use the node name or the node number for id. Terminate the node identification with two colons (::).

Associated Messages

[PLSOPN OPERATOR AT node-id HAS BEEN NOTIFIED AT time]

This message is printed on your terminal when the operator receives a message from your terminal. The node-id is the node where the operator is. The time is when the operator's terminal received the message.

Examples

1. To enter a dialog with the operator, type PLEASE and press RETURN.

```
.PLEASE<RET>
ENTER TEXT, TERMINATE WITH ESCAPE OR ^Z
```

PLEASE responds with the instruction message. In this case, you press <CTRL/Z> after the text because you want to wait for a response from the operator.

```
DO YOU HAVE THE NEW SOS ON LINE YET?<RET>
I CAN'T FIND IT ON SYS<CTRL/Z>
[PLSOPN OPERATOR AT K1514(14) HAS BEEN NOTIFIED AT 14:03:10]
```

The operator responds by saying that your request is finished. PLEASE prompts you with the instruction message. You thank the operator, press <ESC>, and your job returns to monitor level.

```
THE NEWEST VERSION OF SOS IS IN NEW
ENTER TEXT, TERMINATE WITH ESCAPE OR ^Z
THANKS<ESC>
.
```

2. In this example, you send a one-line message to the operator, and you wait for a reply.

```
.PLEASE WHAT TIME IS SYSTEM SHUTDOWN?<RET>
[PLOPSN OPERATOR AT KI514(14) HAS BEEN NOTIFIED AT 14:00:03]

THE SYSTEM WILL GO DOWN AT 6 PM
ENTER TEXT, TERMINATE WITH ESCAPE OR ^Z
THANK YOU<ESC>

.
```

The message from the operator said that the system would shut down at 6:00. You end the dialog by pressing <ESC>.

3. This example shows the use of PLEASE to send a one-line message to the operator at node 26.

```
.PLEASE/NODE:26::ANSWER MY MOUNT REQUEST FOR DSKR:<ESC>
[PLSOPN OPERATOR AT KL1026(26) HAS BEEN NOTIFIED AT 14:30:35]

.
```

PLOT Command

Function

The PLOT command places entries in the plotter output queue. Refer to the QUEUE command for further information and examples.

Format

PLOT dev:jobname=file-spec

Where: **dev** is the name of the specific device on which your files should be plotted. (For example, PLT2: is plotter number 2.) You can have your files plotted on a device connected to another node by using the format PLTSxx: where xx is the node number. (For example, PLTS31 is a plotter on node 31.) The device name is optional.

jobname is the name of the job being entered into the queue. The job name is optional. The default is the name of the first file in the request.

The equal sign is required if you specify either the device name, the job name, or both.

file-spec is a single file specification or a string of file specifications, separated by commas. A file specification is in the form dev:file.ext[directory].

If you specify neither a job name nor a file specification, the system prints a list of the jobs in the plotter queue on your terminal.

The switches to this command can be divided into two categories, depending on whether the switch can be used only once, or can be used more times, in a single command string. The two categories are:

- **Queue-Operation Switches**

  These switches can be used only once in the command string. They affect the entire request, and you can place them anywhere in the command string. If you have used one of these switches in a command string, you cannot use it again in the same string. Many switches have a /NO construction, which has a negative effect. Be sure you do not use the /NO construction of a switch in the same command string with the positive construction.

- **File-Control Switches**

  These switches can be used any number of times in the command string. You can also use the /NO construction of a switch in the same command string with the positive construction. To achieve a temporary or permanent effect by the placement of the switch, refer to Section 1.8.4.

Switch	Category	Function
/ABEFORE: date-time	File control	Queues the file only if the access date is before the specified date and time.
/ACCOUNT: "string"	Queue operation	Specifies the account to which the job should be charged. If the account string contains any nonalphanumeric characters, you must enclose the string in quotation marks.
/AFTER: date-time	Queue operation	Processes the request after the specified date and time.
/ALLFILES: YES or NO	Queue operation	Accepts a request only if all of the files in the request exist. By default, if any of the files do not exist, the others will be processed appropriately. This switch specifies that if any file does not exist, no files should be processed. The value of YES or NO is optional. If you specify YES, all the files you specified must exist or none of the files will be plotted.
/ASINCE: date-time	File control	Queues only the files that have been accessed since the specified date and time.
/BEFORE: date-time	File control	Queues only the files that were created before the specified date and time.
/CHARACTERISTIC: arg	Queue operation	Specifies an output characteristic. You can find a list of the characteristics arguments defined for your system in the file SYS:CHARTY.DAT.
/CHECK	Queue operation	Prints on your terminal a list of the queue entries made by your job.
/COPIES:n	File control	Repeats the output the specified number of times (n must be less than 64). The default is 1 (one copy).
/CREATE	Queue operation	Makes a new entry in the plotter queue. This switch is the default, except when you are listing queue entries.

/DEFERRED	Queue operation	Causes deferred output to be released to the queue. You must use one of the following switches with /DEFERRED: /CREATE completes all released output requests. /KILL eliminates the released output requests. Refer to the SET DEFER command for more information and examples.
/DELETE	File control	Deletes the file after processing it. This is the same as /DISPOSE:DELETE.
/DESTINATION: node	Queue operation	Specifies the node that will process the output. Use this switch to specify that the file should be plotted on a plotter that is connected to the specified node. Use the node name or the node number to specify the node. If a request is made to a node that does not exist, the request will wait in the queue indefinitely.
/DISPOSE:arg	File control	Controls the disposition of the file after it is queued. The arguments to this switch are: DELETE deletes the file from your directory after plotting it. PRESERVE preserves the file after plotting it. This is the default function. RENAME renames the file into the queuing area, deleting it from your directory immediately.
/DISTRIBUTION: "text"	Queue operation	Specifies text to place in the distribution field, on the banner page of output. You can use this field to include mailing information, or the location where the operator should leave the listing. The text field may be up to 39 alphanumeric characters, including punctuation and spaces if the text is placed in quotation marks.
/ERBINARY	File control	Prints an error message if a binary file is included in the request. This is the default function.
/ERNONE	Queue operation	Prints an error message if no files match the file specification. This is the default function.

/ERPROTECTION	Queue operation	Prints an error message if processing the request would require a violation of protection codes. This is the default function.
/FAST	Queue operation	Prints the entries in the queue on your terminal using a fast format.
/FORMS:arg	Queue operation	Specifies any special paper to be used. Available forms are listed in SYS:FORMST.DAT.
/GENERIC	File control	Sends output to the first available plotter. This is the default function. This switch is the complement to /UNIT.
/HEADER: YES or NO	File control	Makes header units before each file if you specify YES. Does not make headers if you specify NO. 1 may be used for YES; 0 may be used for NO. /HEADER:NO is the same as /NOHEADER. /HEADER is the default function.
/HELP:arg	Queue operation	Prints information on your terminal about the QUEUE command. This switch does not queue any files. This switch can be used alone or with one of the following arguments: TEXT prints a message about the format and switches to the QUEUE command. This is the same as /HELP with no arguments. SWITCHES prints a list of all the switches available with the QUEUE command.
/JOBNAME:name	Queue operation	Specifies the name of the job. The job name can be up to six alphanumeric characters.
/KILL	Queue operation	Removes the specified entry from the queue. You must specify the job name, /REQUESTID, or /SEQUENCE, to the left of the equal sign.
/LENGTH:n:m	File control	Processes only files whose length is between n and m blocks.
/LIMIT:n	Queue operation	Limits the output to the specified number of minutes.

/LIST:arg	Queue operation	Prints a list of the jobs in the queue. If you use /LIST alone, it shows the jobs in the queue. This is equivalent to using the QUEUE command with no arguments and no switches. /LIST can be abbreviated to /L. The switch can also take one of the following arguments: ALL shows all data about each queue request. FAST shows a fast list of the queue requests. (This is the same as /FAST.) JOBS shows a list of the jobs in the queue. (This is the same as /LIST with no arguments.) SUMMARY shows only the summary line of the queue display.
/MESSAGE: arg	Queue operation	Specifies the amount of information to be printed when an error occurs from the request. You can specify one or more of the following arguments: ADDRESS prints the location in memory where the error occurred. CONTINUATION prints information about the error. FIRST prints the one-line error message. PREFIX prints a six-character error prefix.
/MODIFY	Queue operation	Alters the specified parameter in the specified job. This switch requires that you have access rights to the job. You must specify a job name, /REQUESTID or /SEQUENCE, to the left of the equal sign in the command line.
/NEW: YES or NO	File control	Accepts the request even if the file does not yet exist.
/NOHEADER	File control	Suppresses header units for each file. /HEADER is the default function.
/NONEW	File control	Does not accept file specifications of files that do not exist. This is the default function.
/NONOTIFY	Queue operation	Suppresses notification of your job when your request is finished. This is the default function. Refer to /NOTIFY.

/NONULL	Queue operation	Prints a fatal error message on a null request. This is the default function.
/NOOPTION	Queue operation	Suppresses the option file SWITCH.INI.
/NOPHYSICAL	File control	Recognizes logical names in the command string. This is the default function.
/NOSTRS	File control	When searching structures for file, uses only the first occurrence. This is the default function.
/NOTES:text	Queue operation	Plots the text in the header columns. The text can be up to 12 characters, and it must be enclosed in quotes if it contains any nonalphanumeric characters, such as spaces.
/NOTIFY:arg	Queue operation	Notifies you on your terminal when your request is completed. To be notified, use /NOTIFY with no argument, or with YES or 1 as an argument. To suppress notification, use /NOTIFY:0, /NOTIFY:NO, or /NONOTIFY. By default, you are not notified when a request is finished. In special cases, such as output of deferred requests, you will never be notified.
/NULL: YES or NO	Queue operation	Does not print a fatal error message if the specified files do not exist. /NULL:NO is the same as /NONULL.
/OKBINARY	File control	Accepts files whose extensions indicate that they include binary information. Normally, files with extensions .SAV, .SHR, .LOW, .REL, .EXE, and .HGH will not be accepted for processing.
/OKNONE	Queue operation	Does not produce an error message if no files match the file specification.
/OKPROTECTION	Queue operation	Does not produce an error message if a file protection code is violated.
/OPTION:option	Queue operation	Uses the option line QUEUE:option in the SWITCH.INI file. SWITCH.INI files are discussed in Appendix B.
/PHYSICAL	File control	Does not recognize logical names in the command line.

/PLOT:arg	File control	Plots the file in the specified mode. If you omit /PLOT, the file is plotted according to the data mode specified in the file. The argument can be any one of the following: ASCII plots the file in ASCII mode. BINARY plots the file in binary mode. IMAGE plots the file in image mode. This is the default function.
/PRESERVE	File control	Saves the file after plotting it. This is the same as /DISPOSE:PRESERVE and it is the default function.
/PRIORITY:n	Queue operation	Gives the specified priority number (n is 1 to 63) to the request. A larger number has greater priority.
/PROTECTION: nnn	Queue operation	Specifies a protection code for the queue request. Queue requests are protected in the same way that files are protected. Refer to Section 1.9.4.
/REMOTE	Queue operation	Prints on your terminal a list of remote queues. Must be used with /DESTINATION.
/REQUESTID:n	Queue operation	Specifies the request identification number of a request you wish to modify or terminate. The request identification number is assigned when the request is queued.
/RUN:file	Queue operation	Executes the specified file after the request is made.
/RUNCORE:n	Queue operation	Executes the file specified /RUN in nK of core after the request is made.
/RUNOFFSET:n	Queue operation	Executes the file specified in /RUN offset n after the request is made.
/SEQUENCE:n	Queue operation	Specifies a sequence number to aid in identifying a request to be modified or deleted.
/SINCE: date-time	File control	Queues only the files with creation dates after the specified date and time.
/STRS: YES or NO	File control	Searches for the file on all structures in the search list and takes every occurrence. The default is to take just the first occurrence of the file.

/TMPFIL: file:text	Queue operation	Creates a temporary file TMP:file and enters the text into the temporary file.
/UNIT:n	Queue operation	Specifies the unit number of the device that you want the output sent to.
/USERNAME: "name"	Queue operation	Specifies the user name field for the banner page of output. This field can contain up to 39 alphanumeric characters, and may include punctuation and spaces if the name is placed in quotation marks.

Associated Messages

When a new entry is made in a system queue, the system prints a message on the user's terminal. The message is in the form:

[PLOTTER JOB name QUEUED, REQUEST #nnn, LIMIT xxx]

Where: **name** is the name of the job in the queue. This can be specified by the user in the PLOT command string. Otherwise, it defaults to the name of the first file in the request.

nnn is the number that represents the request identification of the job in the queue.

xxx is the maximum number of minutes that the job will use.

Characteristics

Leaves your terminal in monitor level.

Destroys your core image.

Does not require LOGIN when you desire a list of the queue entries.

Runs the QUEUE program.

Example

Plot all files with the extension .PLT. The operator is asked to put PLAIN paper on the plotter.

```
.PLOT *.PLT/FORMS:PLAIN<RET>
[PLOTTER JOB MLB QUEUED, REQUEST #172, LIMIT 30]
```

.

POP Command

Function

The POP command returns your job to a superior job context. It reverses the action of a PUSH command, destroying the inferior context and restoring the superior one as the current context. (Section 1.5 contains a discussion of contexts.)

Format

POP

Characteristics

Leaves your terminal at monitor level.

Requires LOGIN.

Destroys the core image of the inferior context and returns you to the preserved context.

Associated Commands

CONTEXT Allows you to display and manipulate contexts.

PUSH Creates an inferior context.

Example

The following CONTEXT command shows that three of the four available contexts are in use at this time, as well as 121 of 1,000 available saved-pages. Use the POP command, and then issue another CONTEXT command, and you can see that one context is freed, some pages are freed, and the current context (indicated by an asterisk (*)) changes from context 3 (BOTLVL) to context 2 (LVLTWO). Context 3 no longer exists.

```
.CONTEXT
Contexts used/quota = 3/4, pages used/quota = 121/1000

Context       Superior      Prog     Idle time
  TOPLVL    1                 MS        :01:36
  LVLTWO    2    TOPLVL    1    PATH      12.46
* BOTLVL    3    LVLTWO    2

.POP

.CONTEXT
Contexts used/quota = 2/4, pages used/quota = 100/1000

Contexts      Superior      Prog     Idle time
  TOPLVL    1                 MS        :01:49
* LVLTWO    2    TOPLVL    1    PATH
```

PRESERVE Command

Function

The PRESERVE command changes the protection code in the owner field of the specified file to 1. 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 1.9.4.

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

NOTE

The actions performed by the PRESERVE command are meaningful **only** to the KJOB and LOGIN CUSPs.

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 - Allows you to specify the protection code for each file or all files. The function of PRESERVE is identical to the function of PROTECT.

Restrictions

The PRESERVE command has no effect on sub-file directories (that is, files with the extension .SFD).

Characteristics

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>   dd-mmm-yy    DSKC:    [27,5434]
TEST    SFD     1  <755>   dd-mmm-yy
TEST    REL     1  <055>   dd-mmm-yy
TEST    MAC     1  <055>   dd-mmm-yy
  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
```

Show that the protection of TEST.REL is now <155>:

```
.DIR TEST.*<RET>

TEST    FOR     1  <055>   dd-mmm-yy    DSKC:    [27,5434]
TEST    SFD     1  <775>   dd-mmm-yy
TEST    REL     1  <155>   dd-mmm-yy
TEST    MAC     1  <055>   dd-mmm-yy
  TOTAL OF 4 BLOCKS IN 4 FILES ON DSKC: [27,5434]
```

PRINT Command

Function

The PRINT command queues files to be printed on the line printer. Refer to the QUEUE command for further information and examples.

Format

PRINT dev:jobname=file-spec

Where: **dev:** is the device name of the specific line printer on which your files are to be printed. (For example, LPT2: is line printer number 2.) You can specify that your files be printed on a printer on another node by using the format LPTSxx, where xx is the node number. (For example, LPTS31 is a printer on node 31.) The device name is optional.

jobname is the name of the job being entered into the queue. The job name is optional. The default is the name of the first file in the request.

The equal sign is required if you specify the device, job name, or both.

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

If you specify neither the job name nor the input specification, the system prints a list of all the jobs in the lineprinter queue on your terminal.

You can use the asterisk wildcard construction for the input specification. Switches that aid in constructing the queue entry can appear as part of the input specifications.

The switches to this command can be divided into two categories, depending on whether the switch can be used only once, or can be used more times, in a single command string. The two categories are:

- **Queue-Operation Switches**

  These switches can be used only once in the command string. They affect the entire request, and you can place them anywhere in the command string. If you have used one of these switches in a command string, you cannot use it again in the same string. Many switches have a /NO construction, which has a negative effect. Be sure you do not use the /NO construction of a switch in the same command string with the positive construction.

- **File-Control Switches**

  These switches can be used any number of times in the command string. You can also use the /NO construction of a switch in the same command string with the positive construction. To achieve a temporary or permanent effect by the placement of the switch, refer to Section 1.8.4.

Switch	Category	Function
/ABEFORE: date-time	File control	Queues the file only if the access date is before the specified date and time.
/ACCOUNT: "string"	Queue operation	Specifies the account to which the job should be charged. If the account string contains any nonalphanumeric characters, you must enclose the string in quotation marks.
/AFTER: date-time	Queue operation	Processes the request after the specified date and time.
/ALLFILES: YES or NO	Queue operation	Accepts a request only if all of the files in the request exist. By default, if any of the files do not exist, the others will be processed appropriately. This switch specifies that if any file does not exist, no files should be processed. The value of YES or NO is optional. If you specify YES, all of the specified files must exist.
/ASINCE: date-time	File control	Queues only the files that have been accessed since the specified date and time.
/BEFORE: date-time	File control	Queues only the files with creation dates before the specified date and time.
/BEGIN:n	File control	Starts the output on the specified page.
/CHARACTERISTIC: arg	Queue operation	Specifies an output characteristic. You can find a list of the characteristics arguments defined for your system in the file SYS:CHARTY.DAT.
/CHECK	Queue operation	Prints on your terminal a list of the queue entries made by your job.
/COPIES:n	File control	Repeats the output the specified number of times (n must be less than 64).
/CREATE	Queue operation	Makes a new entry in the queue. This is the default except when you are listing queue entries.

/DEFERRED	Queue operation	Causes all deferred output to be released to the system queues. You must use one of the following switches with /DEFERRED: /CREATE completes all released output requests. /KILL eliminates the released output requests. Refer to the SET DEFER command for more information and examples.
/DELETE	File control	Deletes the file after printing it. This is the same as /DISPOSE:DELETE
/DESTINATION: node	Queue operation	Specifies the node that will process the request. Use this switch to specify that the files are to be printed on a line printer connected to the specified node. Use the node name or node number to specify the node.
/DISPOSE:arg	File control	Controls the disposition of the file after it is queued. The arguments to this switch are: DELETE deletes the file from your directory after printing it. PRESERVE preserves the file after printing it. This is the default function. RENAME renames the file into the queuing area, deleting it from your directory immediately.
/DISTRIBUTION: "text"	Queue operation	Specifies text to place in the distribution field, on the banner page of output listings. You can use this field to include mailing information, or the location where the operator should leave the listing. The text field may be up to 39 alphanumeric characters, including punctuation and spaces if the text is placed in quotation marks.
/ERBINARY	File control	Prints an error message if a binary file is included in the queue. This is the default function.
/ERNONE	Queue operation	Prints an error message if no files match the file specification. This is the default function.

/ERPROTECTION	Queue operation	Prints an error message if processing the request would require a violation of file protection code. This is the default function.
/FAST	Queue operation	Prints the entries in the queue on your terminal in a fast format.
/FILE: arg	File control	Specifies the way the file is to be interpreted. The following arguments can be used with this switch: ASCII interprets the file as ASCII text. COBOL interprets the file as COBOL SIXBIT text. FORTRAN interprets the file as a Fortran data file. This is the default for files with the extension .DAT.
/FONT: fontname	Queue operation	Prints the file using the font designated in **fontname**. Fontnames vary from installation to installation. See your system administrator for a list of valid font names. (Section 1.16 describes how to print a file using multiple fonts.)
/FORMS:name	Queue operation	Prints the file on the named forms. The argument to the switch can be up to six alphanumeric characters. Available forms are listed in SYS:FORMST.DAT.
/GENERIC	Queue operation	Sends output to the next available line printer. This switch is the complement to /UNIT.
/HEADER: YES or NO	File control	Prints block headers at the beginning of the file. This is the default function.
/HELP: arg	Queue operation	Prints information on your terminal about the QUEUE command. This switch does not queue any files. This switch can be used alone (/HELP) or with one of the following arguments: TEXT prints a message with the format and switches to the QUEUE command. This is the same as /HELP with no arguments. SWITCHES prints a list of all the switches available with the QUEUE command.

/JOBNAME: name	Queue operation	Specifies the name of the job. The job name can be up to six alphanumeric characters.
/KILL	Queue operation	Removes the specified entry from the queue. You must include the job name, /REQUESTID, or /SEQUENCE to the left of the equal sign in the command line. (Refer to the examples.)
/LENGTH:n:m	File control	Prints only files whose length is between n and m blocks.
/LIMIT:n	Queue operation	Limits the output to the specified number of pages.
/LIST:arg	Queue operation	Prints information about the jobs in the queue. If you use /LIST alone, it shows the jobs in the queue. This is equivalent to using the PRINT command with no arguments and no switches. /LIST and /LIST: can be abbreviated to /L and /L:. The switch can also take one of the following arguments: ALL shows all data about each queue request. FAST shows a fast list of the queue requests. JOBS shows a list of the jobs in the queue. (This is the same as /LIST with no arguments.) SUMMARY shows only the summary line of the queue display.
/LOWERCASE	Queue operation	Forces the output to be printed on a line printer with lowercase ability.
/MESSAGE:arg	Queue operation	Specifies the amount of information to be printed when an error occurs from the request. You can specify one or more of the following arguments: ADDRESS prints the location in memory where the error occurred. CONTINUATION prints information about the error. FIRST prints the one-line error message. PREFIX prints the six-character error prefix.

/MODIFY	Queue operation	Alters the specified parameter in the specified job. This switch requires that you have access rights to the job. You must include the job name, /REQUESTID, or /SEQUENCE, to the left of the equal sign in the command line. (Refer to the examples.) You can modify a request as long as the request has not been started.
/NEW: YES or NO	File control	Accepts the request even if the file does not yet exist.
/NOHEADER	Queue operation	Suppresses the block headers at the beginning of the file.
/NONEW	File control	Does not accept file specifications of files that do not exist. This is the default function.
/NONOTIFY	Queue operation	Suppresses notification when your request is finished. Refer to /NOTIFY.
/NONULL	Queue operation	Prints a fatal error message if none of the specified files exist. This is a default function.
/NOOPTION	Queue operation	Suppresses the option file SWITCH.INI.
/NOPHYSICAL	File control	Recognizes logical names for devices in the command string. This is a default function.
/NOSTRS	File control	When searching for the file on all structures, takes only the first occurrence. This is the default function.
/NOTES:"text"	Queue operation	Prints the specified text on the header pages of the output. The text can be 12 characters, and if it contains any nonalphanumeric characters, it must be enclosed in quotation marks.
/NOTIFY: YES or NO	Queue operation	Notifies you on your terminal when your request is completed. To be notified, use /NOTIFY with no argument, or with YES or 1 as an argument. To suppress notification, use /NOTIFY:0, /NOTIFY:NO, or /NONOTIFY. By default, you are not notified when a request is finished. In special cases, such as the printing of log files and the output of deferred requests, you will never be notified.

/NULL: YES or NO	Queue operation	Does not print a fatal error message if the specified files do not exist. /NULL:NO is the same as /NONULL.
/OKBINARY	Queue Operation	Accepts files whose extensions indicate that they include binary information. Normally files with extensions .SAV, .SHR, .LOW, .REL, .EXE, and .HGH will not be accepted.
/OKNONE	Queue operation	Does not produce a warning message if no files match the file specification.
/OKPROTECTION	Queue operation	Does not produce an error message if a file protection code is violated.
/OPTION:option	Queue operation	Uses the option line QUEUE:option in the SWITCH.INI file. SWITCH.INI files are discussed in Appendix B.
/PHYSICAL	File control	Does not recognize logical names in the command line.
/PRESERVE	File control	Saves the file after printing it. This switch is the same as DISPOSE:PRESERVE. This is the default function.
/PRINT:arg	File control	Prints the file with the specified characteristics. The default printing mode is ASCII. The argument can be any one of the following: ARROW prints the file literally, denoting each control character by a ^ and the character, except for the following which are printed literally: carriage return, linefeed, horizontal tab, vertical tab, and formfeed. ASCII prints the file with no changes. GRAPHICS causes the LN01 laser printer to recognize escape sequences. These embedded sequences allow font changes within the file. Unless you specify this switch, escape sequences will be ignored and printed as part of the file. (See Section 1.16 for a discussion of LN01 escape sequences.)

		OCTAL prints the octal values instead of the characters in the file. SUPPRESS converts all control characters to line feeds except for ASCII code characters CR and DC3.
/PRIORITY:n	Queue operation	Gives the specified priority number (n is 1 to 63) to the request. A larger number has greater priority.
/PROTECTION: nnn	Queue operation	Specifies a protection code for the queue request. Queue requests are protected in the same way that files are protected. Refer to Section 1.9.4.
/QUEUE:queue	Queue operation	Specifies the remote VAX queue to receive the output from the request.
/REMOTE	Queue operation	Prints on your terminal a list of remote queues. Must be used with /DESTINATION.
/REPORT:code	File control	Processes COBOL report files by printing every line in the file that begins with the specified code.
/REQUESTID:n	Queue operation	Specifies the request identification number of the job you wish to modify or terminate (/KILL). The request identification number is assigned by the system when the request is made. Place this switch to the left of the equal sign in the command line. (Refer to the examples.)
/RUN:file	Queue operation	Executes the specified file after the request is accepted.
/RUNCORE:n	Queue operation	Executes the file specified in /RUN in nK of core after the request is accepted.
/RUNOFFSET:n	Queue operation	Executes the file specified in /RUN with offset n after the request is accepted.
/SEQUENCE:n	Queue operation	Specifies a sequence number to aid in identifying a request to be modified or deleted. This switch must be used to the left of the equal sign in the command line. (Refer to the examples.)
/SINCE: date-time	File control	Queues only the files with creation dates after the specified date and time.

/SPACING: arg	File control	Prints the files with the specified spacing parameters. The default function is to make no spacing changes to the file, which is the same as /SPACING:SINGLE. You can use the following arguments with this switch: SINGLE prints the file with no spacing changes. DOUBLE prints a blank line between every line of the file. TRIPLE prints two blank lines between every line of the file.
/STRS: YES or NO	File control	Searches for the file on all structures in the search list and takes every occurrence. The default is to take just the first occurrence of the file.
/TMPFIL: file:text	Queue operation	Creates a temporary file TMP:file and enters the text into the file.
/UNIT:n	Queue operation	Specifies a decimal unit number or SIXBIT name of the device you want the output sent to.
/UPPERCASE	Queue operation	Forces the output to be printed on an uppercase-only line printer. This switch is the complement to /LOWERCASE.
/USERNAME: "name"	Queue operation	Specifies the user name for the banner page of output listings. This field can contain up to 39 alphanumeric characters, and may include punctuation and spaces if the name is placed in quotation marks.

Associated Messages

When a new entry is made in a system queue, the system prints a message on the user's terminal. The message is in the form:

[PRINTER JOB name QUEUED, REQUEST #nnn, LIMIT xxx]

Where: **name** is the name of the job in the queue. This can be specified by the user. Otherwise, it defaults to the name of the first file in the request.

nnn is the number that represents the request identification of the job in the queue.

xxx is the maximum number of pages that the job will use.

Characteristics

Leaves your terminal in monitor mode.

Destroys your core image.

Does not require LOGIN if you want only a list of jobs in the queue to be printed on your terminal.

Examples

1. Print the file SYS:NOTICE.TXT.

```
.PRINT SYS:NOTICE.TXT<RET>
[PRINTER JOB NOTICE QUEUED, REQUEST #109, LIMIT 50]
.
```

2. Print the file SYS:NOTICE.TXT after 5:00 P.M.

```
.PRINT SYS:NOTICE.TXT /AFTER:17:00<RET>
[PRINTER JOB NOTICE QUEUED, REQUEST #109, LIMIT 50]
.
```

3. Change the job with request identifcation number 109 to print after 4:30 P.M.

```
.PRINT /REQUESTID:109= /MODIFY /AFTER:16:30<RET>
```

4. Cancel the job with sequence number 22.

```
.PRINT /SEQUENCE:22= /KILL<RET>
```

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 1.9.4 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 1.9.

<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 Changes the protection codes of the specified files to <1nn>.

Characteristics

Leaves your terminal at monitor level.

Destroys your core image.

Example

DIRECTORY shows that TEST.REL has the protection <155>:

```
.DIR TEST.*<RET>

TEST    FOR     1  <055>   dd-mmm-yy    DSKC:    [27,5434]
TEST    SFD     1  <775>   dd-mmm-yy
TEST    REL     1  <155>   dd-mmm-yy
TEST    MAC     1  <055>   dd-mmm-yy
  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>   dd-mmm-yy    DSKC:    [27,5434]
TEST    SFD     1  <775>   dd-mmm-yy
TEST    REL     1  <555>   dd-mmm-yy
TEST    MAC     1  <055>   dd-mmm-yy

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

PUNCH Command

Function

The PUNCH command is defined by the system administrator to be equivalent to either the TPUNCH or CPUNCH command. TOPS-10 is shipped with PUNCH defined as TPUNCH, but this may be changed at your site.

Refer to the TPUNCH and CPUNCH commands for further information.

PUSH Command

Function

The PUSH command creates an inferior context; the new context becomes your current one. The core image of the superior context is preserved. (Refer to Section 1.5 for a discussion of contexts.)

Format

PUSH

Characteristics

Leaves your terminal at monitor level.

Requires LOGIN.

Preserves your core image.

Associated Commands

CONTEXT Allows you to display and manipulate contexts.

POP Allows you to return to a superior context.

Example

Run a program, for example, MS. Then exit and issue a CONTEXT command to see the status of your current context.

```
.CONTEXT

Contexts used/quota = 1/4, pages used/quota = 4/1000
Context      Superior   Prog   Idle time
*        1              MS
```

Now issue a PUSH command, and try another CONTEXT command to see the new status of your contexts.

```
.PUSH

.CONTEXT
Contexts used/quota = 2/4, pages used/quota = 6/1000
Context      Superior   Prog   Idle Time
         1              MS      19.78
*        2         1
```

There are now two contexts in use. The current, unnamed context 2 is not running a program. MS, which is running in the unnamed superior context, has been idle for 19.78 seconds since the last CONTEXT command was issued.

QUEUE Command

Function

The QUEUE command makes entries in any of several system queues, or in the case of the Event Queue, displays information only. The system queues are:

- **Event Queue** - Contains information about scheduled system events. This information may include such things as the time the system will go down (KSYS) and when certain files will be closed. This queue is a display queue only.
- **Input Queue** - Contains the batch jobs that have been submitted to the system.
- **Mount Queue** - Contains the requests for resources to be mounted. These are: tape, disk, and DECtape mount requests.
- **Output Queues:**
  - **Card Punch Queue** - Contains the requests that have been made for files to be punched on cards.
  - **Paper Tape Punch Queue** - Contains the requests that have been made for files to be punched on paper tape.
  - **Plotter Queue** - Contains the requests that have been made for files to be plotted.
  - **Printer Queue** - Contains the requests that have been made for files to be printed on a line printer.

When queueing a request, you can specify an individual device. If you do not specify a device, the first available appropriate device is used.

All requests are placed in a queue (a list of the requests, arranged according to priority). The requests are processed in this order. Priority is established by the limit that is set on each request (refer to /LIMIT and /TIME below) and according to the specified priority number (refer to /PRIORITY below). In general, requests with smaller limits and higher priorities are processed first.

The QUEUE command also creates lists of the requests in the queues. This list can be printed on the terminal or entered into a file. The list includes the mount queue as well as input and output queues.

The QUEUE command is duplicated by the CPUNCH, PLOT, PRINT, PUNCH, SUBMIT, and TPUNCH commands. Each of these commands performs a function with a specific system queue.

Formats

1. To process a batch job, use the following format:

   QUEUE INP:jobname=control-file,log-file

   Where: **jobname** is the name of the job. This name must be one to six alphanumeric characters. The job name is optional. If you do not specify the job name, the default is the name of the log file.

   The equal sign is required.

   control-file is the file specification of the batch control file. The file name is required. If you do not specify the file extension, the default is .CTL.

   log-file is the file specification that you want assigned to the log file, which contains a record of the processing of the batch job. The log file specification is optional. If you omit the file name, the default is the name of the control file. If you omit the file extension, the default is .LOG.

2. To make an entry in one of the output queues, use the following format:

   QUEUE dev:jobname=file-spec,file-spec

   Where: **dev**: is the name of the device on which the output is to be processed. The device name may be a generic device name, in which case the first available device is used, or it may be a specific device name. (For example, LPT: is the first available line printer. LPT2: is line printer number 2.) To use a device on another node, use the format devSxx:, where dev is a generic device name, and xx is a node number. (For example, LPTS34: is a line printer on node 34.) If you do not specify a device name, the default is LPT:. The generic device names are:

   LPT: for line printer requests.
   CDP: for card punch requests.
   PLT: for plotter requests.
   PTP: for paper tape punch requests

   jobname is the name of the job. The job name is one to six alphanumeric characters. The job name is optional. If you omit the job name, the name of the first file in the request is used as the job name. However, if the first file does not exist (see /NEW), the name of the second file is used.

   file-spec is the file specification of the file to be processed. The file name and extension are required.

3. To obtain a file containing a list of all the entries in all the queues, use the following format:

   QUEUE file-spec=/LIST

   Where: **file-spec** is the file specification of the output file. If you omit the file extension, the default is .LSQ.

4. To display the entries in any or all system queues on your terminal, use the following format:

   QUEUE dev:

   Where: **dev** is any of the system queue names. These are:

   INP: for the batch input queue.
   CDP: for the card punch queue.
   PLT: for the plotter queue.
   LPT: for the line printer queue.
   PTP: for the paper tape punch queue.

   The device name is optional. If you do not specify the device name, all the entries in all the system queues are printed on your terminal. This list includes the mount queue.

Switches

The switches for the QUEUE, CPUNCH, PLOT, PRINT, PUNCH, SUBMIT, and TPUNCH commands can be placed anywhere in the command line.

The switches can be divided into two categories, depending on whether the switch can be used only once, or can be used more times, in the command line. The two categories are:

- **Queue-Operation Switches**

  These switches can be used only once in the command string. They affect the entire request, and you can place them anywhere in the command string. If you have used one of these switches in a command string, you cannot use it again in the same string. Many switches have a /NO construction, which takes a negative effect. Be sure you do not use the /NO construction of one of these switches in the same command string with the positive construction of the switch.

- **File-Control Switches**

  These switches can be used any number of times in the command string. You can use the /NO construction of one of these switches in the same command string with the positive construction of the switch. For information about how to achieve a temporary or permanent effect by the placement of the switch, refer to Section 1.8.4.

In the following table of switches, the switches are defined according to the queues that they affect. The following labels signify the queue(s) that each switch affects. The switches are defined as follows:

ALL - Switches that affect both the batch input queue and the output queues.

INPUT - Switches that affect only the batch input queue.

OUTPUT - Switches that affect only the output queues. OUTPUT is followed by (queue) in cases where a switch is useful for that specific output queue.

LIST - Switches that affect the listing of the jobs in the queue. Listings can be printed on the terminal, or they can be written into files on disk or tape. LIST is followed by (queue) when the listing function is useful for that queue.

Switch	Category	Queues	Meaning
/ABEFORE: date-time	File control	ALL	Queues the file only if the access date is before the specified date and time.
/ACCOUNT: "string"	Queue operation	ALL	Specifies the account to which the job should be charged. If the account string contains any nonalphanumeric characters, you must enclose the string in quotation marks.
/AFTER: date-time	Queue operation	ALL	Processes the request after the specified date and time.
/ALLFILES: YES or NO	Queue operation	OUTPUT	Accepts a request only if all of the files in the request exist. By default, if any of the files do not exist, the others will be processed appropriately. This switch specifies that if any file does not exist, no files should be processed. The value of YES or NO is optional. If you use YES, all of the specified files must exist. If NO, existing files are processed and warning messages are printed for files that do not exist.

/ASINCE: date-time	File control	ALL	Queues only the files that have been accessed since the specified date and time.
/ASSISTANCE: YES or NO	Queue operation	INPUT	Specifies whether the job needs or does not need operator intervention. Arguments are YES or 1 and NO or 0. If you specify NO, and then request assistance, your job is cancelled. Assistance is any request that the operator must answer before the batch job can continue, including PLEASE and MOUNT requests.
/BATLOG:arg	Queue operation	INPUT	Controls the way the log file is output to disk. Arguments are: APPEND appends the log file to an existing file of the same name. This is the default function. SUPERSEDE replaces any file of the same name with the new log file. SPOOL spools the log file for output. Does not store the file in your directory.
/BATOPT: option-name	Queue operation	INPUT	Specifies a LOGIN option line to read for LOGIN switches to apply to the batch job. The option name that you specify with the /BATOPT switch must match a line in the SWITCH.INI file that appears as: LOGIN:option-name/switches
/BEFORE: date-time	File control	ALL	Queues only the files with creation dates before the specified date and time.
/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.

/CARDS:n	Queue operation	INPUT	Uses n as the maximum number of cards that can be punched by your batch job (up to 10,000).
/CHARAC:arg	Queue operation	OUTPUT	Specifies an output characteristic. The argument can be up to six alphanumeric characters. Characteristics arguments defined for your system are listed in the file SYS:CHARTY.DAT.
/CHECK	Queue operation	LIST	Prints on your terminal a list of the queue entries made by your job.
/COPIES:n	File control	OUTPUT	Repeats the output the specified number of times (n must be less than 64).
/CORE:n	Queue operation	INPUT	Uses n (in decimal K) as the maximum amount of memory that your job can use.
/CREATE	Queue operation	ALL	Makes a new entry in the queue. This switch is the default except when you are listing queue entries.
/DEFERRED	Queue operation	OUTPUT	Causes all deferred output to be released to the system queues. You must use one of the following switches with /DEFERRED: /CREATE completes all released output requests. /KILL eliminates the released output requests. Refer to the SET DEFER command for more information and examples.
/DELETE	File control	ALL	Deletes the file after processing it. (This is the same as /DISPOSE:DELETE.)
/DENSITY:n	Queue operation	LIST	Specifies the density of magnetic tape when writing listing files directly on tape. The values for n are: 200, 800, 1600, 6250, and INSTALLATION (to take the installation default).

/DEPENDENCY:n	Queue operation	INPUT	Specifies the initial value of the dependency count in decimal. When used with /MODIFY, this switch changes the dependency count of another job. If n is a signed number (+ or -), that number is added to or subtracted from the dependent job's count. If n is not a signed number, the dependent job's count is changed to n.
/DESTINATION: node	Queue operation	ALL	Specifies the node to which all output is to be sent. When used with an output request, the file will be output on a device attached to the specified node. When used with an input request, any output generated by the request (including the log file, if it is to be printed) is sent to the specified node. The node can be specified by either the node name or the node number. If a request is made to a node that does not exist, the request will wait in the queue indefinitely.
/DISPOSE: arg	File control	ALL	Controls the disposition of the file after it is queued. The arguments to this switch are: DELETE deletes the file from your directory after processing it. PRESERVE preserves the file after processing it. This is the default function. RENAME renames the file from your area into the spooling area, deleting it from your area immediately.

/DISTRIBUTION: "text"	Queue operation	ALL	Specifies text to place in the distribution field, on the banner page of output listings. For batch input requests, the distribution text is printed on the banner page of the log file listing. You can use this field to include mailing information, or the location where the operator should leave the listing. The text field may be up to 39 alphanumeric characters, including punctuation and spaces if the text is placed in quotation marks.
/ERBINARY	File control	ALL	Prints an error message if a binary file is included in the request. This is the default function.
/ERNONE	Queue operation	ALL	Prints an error message if no files match the file specification. This is the default function.
/ERPROTECTION	Queue operation	ALL	Prints an error message if processing the request involves a violation of file protection. This is the default function.
/ERSUPERSEDE	Queue operation	LIST	Prints an error message if a listing file by the same name already exists. Without this switch, the old listing file is automatically superseded by the new listing file.
/ESTIMATE:x	Queue operation	LIST	Specifies an estimated number of blocks for the listing file.
/FAST	Queue operation	LIST	Prints the entries in the queue on your terminal in a fast format. Same as /LIST:FAST.
/FEET:n	Queue operation	INPUT	Uses n as the maximum number of feet of paper tape that your batch job can punch.

/FILE:arg	File control	OUTPUT	Specifies how the file format is to be interpreted. The following arguments can be used with this switch: ASCII interprets the file as ASCII text. COBOL interprets the file as COBOL SIXBIT text. ELEVEN interprets the file as four 8-bit bytes in each 36-bit word. The bits are arranged as follows: Byte 1: bits 10-17 Byte 2: bits 2-9 Byte 3: bits 28-35 Byte 4: bits 20-27 FORTRAN interprets the file as a FORTRAN data file. This is the default for files with the extension .DAT.
/FONT:name	Queue operation	OUTPUT	Specifies that the file should be printed entirely in the font designated in name. As fontnames vary from installation to installation, see your system administrator for a list of valid fontnames.
/FORMS:arg	Queue operation	OUTPUT	Processes the output on the specified type of form. The argument to the switch can be up to six alphanumeric characters. Available forms are listed in SYS:FORMST.DAT.
/GENERIC	Queue operation	OUTPUT	Sends output to any unit of the type specified or implied. For instance, if no device is specified, output goes to the next available line printer. This switch is the complement to /UNIT.
/HEADER: YES or NO	Queue operation	OUTPUT	Prints block headers between copies of the file. This is the default function.

/HELP: arg	Queue operation		Prints information on your terminal about the QUEUE command. This switch does not queue any files. This switch can be used alone or with one of the following arguments: TEXT prints a message about the format and switches of the QUEUE command. This is the same as /HELP with no arguments. SWITCHES prints a list of all the switches available with the QUEUE command.
/JOBNAME:name	Queue operation	ALL	Specifies the name of the job. The job name can be up to six alphanumeric characters.
/KILL	Queue operation	ALL	Removes the specified entry from the specified queue. You must include the job name, /SEQUENCE, or /REQUESTID to the left of the equal sign in the command line. Refer to the examples.
/LENGTH:n:m	File control	ALL	Processes only files whose length is between n and m blocks.
/LIMIT:n	Queue operation	OUTPUT	Limits the output to the specified number of pages, cards, feet, or minutes.
/LIST: arg	Queue operation	LIST	Prints information about the jobs in the queue. If you use /LIST alone, it shows the jobs in the queue. This is equivalent to using the QUEUE command with no arguments or switches. /LIST can be abbreviated to /L. The switch can also take one of the following arguments: ALL shows all data about each queue request. FAST shows a fast list of the queue requests.

			JOBS shows a list of the jobs in the queue. (This is the same as /LIST with no arguments.) SUMMARY shows only the summary line of the queue display.
/LOWERCASE	Queue operation	OUTPUT (LPT)	Forces the output to be printed on a line printer with lowercase ability.
/MESSAGE: arg	Queue operation	ALL	Specifies the amount of information to be printed when an error occurs from the request. You can specify one or more of the following arguments: ADDRESS prints the location in memory where the error occurred. CONTINUATION prints information about the error. FIRST prints the one-line error message. PREFIX prints a six-character error prefix.
/METERS:n	Queue operation	INPUT	Uses n as the maximum number of meters of paper tape that can be punched by the batch job.
/MODIFY	Queue operation	ALL	Alters the specified parameter in the specified job. This switch requires that you have access rights to the job. You must include the job name, /REQUESTID, or /SEQUENCE, to the left of the equal sign in the command line. (Refer to the examples.) This switch can be used to modify a previously submitted request as long as the request has not been started.
/NEW: YES or NO	File control	ALL	Accepts the request even if the file does not yet exist.
/NOHEADER	Queue operation	OUTPUT	Suppresses the block headers at the beginning of the file.

/NONEW	File control	ALL	Does not accept file specifications of files that do not exist. This is the default function.
/NONOTIFY	Queue operation	ALL	Does not notify you when when your request is finished. This is the default function. Refer to /NOTIFY.
/NONULL	Queue operation	OUTPUT	Prints a fatal message on a null request. This is the default function.
/NOOPTION	Queue operation	ALL	Does not use the parameters specified in the SWITCH.INI file.
/NOPHYSICAL	File control	ALL	Recognized logical names for devices in the command string. This is the default function.
/NORESTART	Queue operation	INPUT	Prevents your job from being restarted when it was stopped because of an error.
/NOSTRS	File control	OUTPUT	When searching structures for the specified file, uses only the first file found. This is the default function.
/NOTES:"text"	Queue operation	OUTPUT	Includes the text in the header units of output. you should include quotation marks if the text contains spaces or punctuation.
/NOTIFY:arg	Queue operation	ALL	Tells the system whether to send a message to your terminal when your request is completed. By default, you are not notified when a request is finished. To be notified, use /NOTIFY with no argument, or with YES or 1 as an argument. To suppress notification, use /NOTIFY:0, /NOTIFY:NO, or /NONOTIFY. In special cases, such as the printing of log files and the output of deferred requests, you will never be notified.

/NULL: YES or NO	Queue operation	OUTPUT	Does not print a fatal error message if the files you specified do not exist. /NULL:NO is the same as /NONULL.
/OKBINARY	File control	ALL	Accepts files whose extensions indicate that they include binary information. Normally files with extensions .SAV, .SHR, .LOW, .REL, .EXE and .HGH will not be accepted.
/OKNONE	Queue operation	ALL	Does not produce a warning message if none of the specified files exist.
/OKPROTECTION	Queue operation	ALL	Does not generate an error message if a file protection code is violated.
/OKSUPERSEDE	Queue operation	LIST	Does not generate an error if the listing file already exists. This is the default function.
/OPTION:option	Queue operation	ALL	Uses the line QUEUE: option in the SWITCH.INI file. SWITCH.INI files are discussed in Appendix B.
/OUTPUT:arg	Queue operation	INPUT	Determines whether or not the log file will be printed. 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.
/PAGES:n	Queue operation	INPUT	Uses n as the maximum number of pages of output that your job can print.
/PARITY: ODD or EVEN	Queue operation	LIST	Uses the specified parity, when writing the listing file directly onto magnetic tape. The default is ODD.
/PATH:[dir]	Queue operation	INPUT	Specifies the directory to be accessed by the job. The directory can be UFD or an SFD.

/PHYSICAL	File control	ALL	Does not recognize logical names for devices in the command line.
/PLOT:arg	File control	OUTPUT (PLT)	Plots the file in the specified mode. If you omit /PLOT, the file is plotted according to the data mode specified in the file. The argument can be any one of the following: ASCII plots the file in ASCII mode. BINARY plots the file in binary mode. IMAGE plots the file in image mode.
/PRESERVE	File	ALL	Saves the file after <le>.I-24 control processing it. (This is the same as /DISPOSE:PRESERVE). This is the default function.
/PRINT:arg	File control	OUTPUT (LPT)	Prints the file using the argument to determine the printing mode. The default printing mode is ASCII. The argument can be any one of the following: ARROW converts all control characters except 011-015 and 020-024 to up-arrow format. ASCII prints the file with no changes. This is the default. GRAPHICS causes the LN01 laser printer to recognize escape sequences. These embedded sequences allow font changes within a file. Unless this switch is specified, escape sequences are ignored and printed as part of the file. (See Section 1.16 for a discussion of LN01 escape sequences.) OCTAL prints the octal values instead of the characters in the file.

			SUPPRESS converts all control characters to line feeds except for ASCII code characters CR and DC3.
/PRIORITY:n	Queue operation	ALL	Assigns the specified priority (n is 1 to 63) to the request. A larger number has greater priority.
/PROCESSING: node	Queue operation	INPUT	Specifies the node that will process the job. You must specify the node by the node name or node number. Batch jobs can be submitted only to IBM host nodes. Jobs that are submitted to nodes other than IBM host nodes will wait in the queue indefinitely.
/PROTECTION: nnn	Queue operation	LIST	Specifies a protection code for the listing file.
		OUTPUT INPUT	Specifies a protection for the queue request. Queue requests are protected in the same way that files are protected. Refer to Section 1.9.4.
/PUNCH:arg	File control	OUTPUT (CDP)	Punches the files in the specified mode. If you omit this switch, the files are punched according to the data mode specified in the file. The following arguments can be used with this switch: 026 punches the files in 026 Hollerith code. ASCII punches the files in ASCII card code. BCD punches the files in 026 Hollerith code. BINARY punches the files in checksummed binary card code. IMAGE punches the files in image card format.
/QUEUE:queue	Queue operation	OUTPUT	Specifies the remote VAX queue to receive the output from the request.

/READER	Queue operation	INPUT	Causes a disk-resident card job to be read as if it were on punched cards and had been submitted through the card reader.
/REMOTE	Queue operation	ALL	Prints on your terminal a list of remote queues. Must be used with /DESTINATION.
/REPORT:code	File control	OUTPUT (LPT)	Prints COBOL reports by printing every line in the file that has the specified code at the beginning of the line.
/REQUESTID:n	Queue operation	ALL	Specifies the request identification number of a job you wish to modify (/MODIFY) or terminate (/KILL). The request identification number is assigned by the system when you queue the request. This switch must be used to the left of the equal sign in the command line. (Refer to the examples.)
/RESTARTABLE: YES or NO	Queue operation	INPUT	Specifies whether the job should be restarted after the system has crashed and been restored. Arguments are: YES or 1, and NO or 0. The default is NO.
/RUN:file	Queue operation	ALL	Executes the specified file after the request is queued.
/RUNCORE:nx	Queue operation	ALL	Executes the program specified in /RUN in nK of core after the request is accepted. The value can also be expressed in terms of nP (pages).
/RUNOFFSET:n	Queue operation	ALL	Executes the file specified in /RUN with offset n after the request is queued.
/SEQUENCE:n	Queue operation	ALL	Specifies the sequence number of a request to be modified or deleted. This switch must be used to the left of the equal sign in the command line.

/SINCE: date-time	File control	ALL	Queues only the files with creation dates after the specified date and time.
/SITGO	Queue operation	INPUT	Specifies that the batch job be processed by the SITGO processor.
/SPACING:arg	File control	OUTPUT (LPT)	Prints the files with the specified spacing parameters. The default function is to make no spacing changes to the file, this is the same as /SPACING:SINGLE. You can use the following arguments with this switch: SINGLE prints the file with no spacing changes. DOUBLE prints a blank line between every line of the file. TRIPLE prints two blank lines between every line of the file.
/STREAM:n	Queue operation	LIST (INP)	Prints a list of the jobs that are running or destined to run in the specified batch stream.
/STRS: YES or NO	File control	OUTPUT	Searches for the file on all structures in the search list and takes every 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.

/TAPE:arg	File control	OUTPUT (PTP)	Punches paper tape in the specified code. If you do not use this switch, the tape is punched according to the data mode specified in the file. You can use any one of the following arguments with this switch: ASCII punches the tape in ASCII code. BINARY punches the tape in binary code. IBINARY punches the tape in image binary code. IMAGE punches the tape in image mode.
/TIME: hh:mm:ss	Queue operation	INPUT	Specifies the CPU time limit for the job. The form /TIME:n can be used to specify a limit of n seconds.
/TMPFIL: file:"text"	Queue operation	ALL	Creates a temporary file TMP:file.TMP and enters the text into the file.
/TPLOT:n	Queue operation	INPUT	Uses n minutes as the maximum amount of plotting time allowed for your job.
/UNIQUE: YES or NO	Queue operation	INPUT	Specifies whether more than one batch job can run from your PPN at one time. If the value is YES (or 1), only one job will run at a time. Any other batch jobs will wait until the previous job is finished. If the value is NO (or 0), any number of batch jobs can run at the same time.
/UNIT:n	Queue operation	OUTPUT	Specifies the unit number or SIXBIT name of the device that you want the output sent to.
/UPPERCASE	Queue operation	OUTPUT (LPT)	Forces the output to be printed on an uppercase-only line printer. This switch is the complement to /LOWERCASE.

/USERNAME: "name"	Queue operation	ALL	Specifies the user name field for the banner page of output listings. For batch input requests, the user name is printed on the banner page for the log file listing. This field can contain up to 39 alphanumeric characters, and may include punctuation and spaces if the name is placed in quotation marks. To avoid conflict with the MOUNT/USER switch, do not shorten this switch to less than /USERN.
/VERSION:n	Queue operation	LIST	Specifies a version number in standard DECsystem-10 format for listing files.

Associated Messages

When a new entry is made in a system queue, the system prints a message on the user's terminal. The message is in the form:

[device JOB name QUEUED, REQUEST #nnn, LIMIT xxx]

Where: **device** is the name of one of the output devices, or BATCH.

name is the name of the job in the queue. This can be specified by the user. Otherwise, it defaults to the name of the first file in the request. In the case of batch requests, the name of the log file is used as the job name.

nnn is the number that represents the request identification of the job in the queue.

xxx is the maximum number of pages that the job will use.

When the entry has been made to the batch queue, the LIMIT is changed to RUN TIME. In this case, the time limit of the job is displayed.

Characteristics

Leaves your terminal in monitor level.

Destroys your core image.

Does not require LOGIN if you desire only a list of queue entries.

Examples

1. Enter file FILEA.LST and FILEB.LST in the line printer queue under the job name of FILEA.

```
.QUEUE FILEA,FILEB<RET>
[PRINTER JOB FILEA QUEUED, REQUEST #4, LIMIT 20]

.
```

2. Enter file TEST.CTL in the batch input queue under job name TEST and log file with name TEST.LOG.

```
.QUEUE INP:=TEST<RET>
[BATCH JOB TEST QUEUED, REQUEST #33, LIMIT 00:05:00]

.
```

3. Submit the control file TEST.CTL to the batch input queue for processing after 5:00 P.M.:

```
.QUEUE INP:  =TEST /AFTER:17:00
[BATCH JOB TEST QUEUED, REQUEST  231, LIMIT 0:05:00]
```

4. Modify the processing time of job TEST, identifying the job by its request number:

```
.QUEUE INP:  /REQUESTID:231= /AFTER:16:30 /MODIFY
[1 JOB MODIFIED]
```

5. Cancel the batch job TEST:

```
.QUEUE INP:  TEST= /KILL
[1 JOB CANCELED]

.
```

6. Display a list of all the entries in all the queues:
   .QUEUE<RET>

```
Event Queue:
Type       Req#       Expiration           Description
--------  -------  ------------------  ---------------------
*BILCLS       9   dd-mmm-yy  hh:mm:ss  Prime time rates end
USGFIL        6   dd-mmm-yy  hh:mm:ss  Usage file closure
OPRFIL        7   dd-mmm-yy  hh:mm:ss  ORION log file closure
BILCLS        8   dd-mmm-yy  hh:mm:ss  Discount rates end
There are 4 events in the queue (1 in progress)

Batch Queue:
Job Name   Req#   Run Time    Core             User
--------  -----   --------   -----   ------------------------
MCOFIL      12  00:05:00     512   SPIDER [30,5653]
/After: dd-mmm-yy  hh:mm
SJH         10  00:04:00     512   SPIDER [30,5653]
/After: dd-mmm-yy  hh:mm
CHKUSR      22  00:05:00     512   LEO    [10,6056]
/After: dd-mmm-yy  hh:mm
MAIL        24  00:15:00     512   NED    [30,5674]
/After: dd-mmm-yy  hh:mm
There are 4 jobs in the queue (none in progress);
00:29:00 runtime

Printer Queue:
Job Name   Req#   Limit             User
--------  -----  -------  ----------------------
* LLINKS   588    1798    DAVENPORT [10,6026]   On Unit:0
/Dest:JUNIPR
Started at 16:31:09
RSX20F     221     924    M.J.MAROTTA [27,5434]  /Lower
There is 2 jobs in the queue (1 in progress); 2722 pages
```

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/switch
```

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 1.14.

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.

/switch is one of the following options:

/START:n,,addr	Specifies the octal section number (n) and address (addr) at which the core image starts. Valid section numbers are 0-37, octal.
/USE:n	Specifies the octal section number (n) in which the core image runs.

Characteristics

Destroys your core image.

Example

```
Run the SYS:SETSRC.EXE program

.R SETSRC<RET>

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 1.9.1 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

Leaves your terminal at monitor level.

Will not function during device I/O.

Restrictions

Your job's controlling terminal cannot be reassigned.

Disk and spooled devices cannot be reassigned.

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 command 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 allow access to the system.

Formats

```
REATTACH
REATTACH terminal-name/switch
REATTACH line-number/switch
```

Where: **terminal-name** is a device name of the form TTYnnn[:].

line-number is the **nnn** portion of the terminal-name.

/switch is optional. The only switch for REATTACH is:

/HELP:keyword Prints the HELP file. Valid keywords are ARGUMENTS, SWITCHES, and TEXT. The ARGUMENTS keyword displays a list of valid switches and arguments. The SWITCHES keyword displays only a list of switches without detailed information. The TEXT keyword displays the full HELP text. TEXT is the default keyword. /HELP may be abbreviated to /H.

Characteristics

Does not destroy the core image of either job.

Requires LOGIN.

Restrictions

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

Batch programs cannot use this command.

Example

Enter the command REATTACH.

.REATTACH TTY56<RET> .

If you do not enter the terminal-name or line-number, the system will prompt you.

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 (PC) value 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

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 1.9 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

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>   dd-mmm-yy     DSKC:[27,5434]
```

Rename PROG to be PRGRM.

```
.RENAME PRGRM.EXE=PROG.EXE<RET>
FILES RENAMED:
DSKC:PROG.EXE
```

Show the file named PRGRM.

```
.DIR PRGRM.*<RET>

PRGRM   EXE     68      <055>   dd-mmm-yy     DSKC:[27,5434]
```

RESOURCES Command

Function

The RESOURCES command prints the names of all available devices (except terminals and pseudo-terminals), all file structures, all physical units not in file structures, and all running CPUs.

Format

```
RESOURCES
```

Characteristics

Leaves your terminal at monitor level.

Example

```
.RESOURCES<RET>

BLKX,RENG,DSKA,BLKB,DSKP,AP10,DSKC,BLKK,DSKR,DSKZ,LOKI,DSKT,
SLAT,SIXT,DSKB,BLKY,RPA2,RPA4,RPB3,RPB6,RPP2,RPP3,RPG1,LPT261,
DTA0,1,2,3,MTA0,1,2,4,5,MTE1,MTF0,1,2,MTC0,MTD0,1,CPU0,1

.
```

REWIND Command

Function

The REWIND command rewinds a magnetic tape or a DECtape.

Format

REWIND dev:

Where: **dev:** is a magnetic tape (MTxn) or a DECtape (DTxn).

Refer to Section 1.9.1 for a description of device names.

Characteristics

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 MTA1:<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/switch

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 memory if this amount is different from either the minimum core needed to load the program, or the core argument of the SAVE command that saved the file.

If core is greater than the minimum low segment size and is less than 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.

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.

/switch is one of the following options:

/START:n,,addr	Specifies the octal section number (n) and address (addr) at which the core image starts. Valid section numbers are 0-37, octal.
/USE:n	Specifies the octal section number (n) in which a core image executes.

Characteristics

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

Function

The SAVE and SSAVE commands create executable image files from programs that are loaded in core memory. 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.

Format

SAVE dev:file.EXE[directory] core/switch

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 file extension. 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.

/switch is the following option:

/START:n,,addr Allows you to save a multi-section program with a starting address in section n, address addr.

Characteristics

Leaves your terminal at monitor level.

Does not operate while I/O is in progress.

Requires memory.

Can save multiple high segments. (SSAVE cannot.)

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>    dd-mmm-yy    DSKC:  [27,5434]
PROG         REL      1 <055>    dd-mmm-yy
PROG         EXE     68 <055>    dd-mmm-yy
  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.
0010	No remote terminals.
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

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

.
```

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 and Packed Image Mode (PIM) 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.

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:

OPRxx:

Where: **xx** is the number of the operator's node.

See Example 4 below.

Characteristics

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>
.
```

SESSION Command

Function

The SESSION command allows you to change the account and remark strings of your job. If your system is not running with account validation, the SESSION command has no effect. If your system is running with account validation, you can use the SESSION command to change the billing profile of your job.

For example, you may have several projects, each of which is charged to a separate account. Therefore, several account strings are valid for your PPN. The SESSION command allows you to change the account string for your job without logging out and logging in again. Thus, all subsequent charges will be made to the new account.

If account validation is enabled, the string you type must match one of the accounts assigned to your PPN. The string must match in case (lower or uppercase), although this feature can be changed by the system administrator.

Formats

```
SESSION<RET>
ACCOUNT:string<RET>
REMARK:text
```

This format prompts you for the required information. If either string is not required, you are not prompted for it.

Where: **string** is the account string for the account to which you want any subsequent charges to be made.

text is the remark string. You can type anything here to identify your work. The remark string is limited to 39 characters.

SESSION/switches

This format allows you to specify one or both strings in the command line. The valid switches are:

/ACCOUNT:"string"	Specifies the account string for the account to which all subsequent charges should be made. The account string must be enclosed in quotation marks if it contains any nonalphanumeric characters. The account string is case sensitive.
/HELP	Prints the HELP text, listing available switches and how to use them.
/REMARK:"text"	Specifies the remark string for subsequent charges. You can type anything for the string ("text") up to 39 characters. The remark string must be enclosed in quotation marks if it contains any nonalphanumeric characters.

Characteristics

Leaves your terminal at monitor level when all required information is supplied.

Destroys your core image.

Examples

1. The following example shows the procedure for changing the account string to DATA-AQ. In this case, the account string is required for the job, but the remark string is not required.

```
.SESSION<RET>
ACCOUNT:DATA-AQ<RET>

.
```

2. The following example shows the procedure for supplying both the account and remark strings in the command line:

```
.SESSION/ACCOUNT:"DATA-AQ"/REMARK:"SALES RESEARCH"<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 MTxn:, or the logical name previously associated with the device. The device must be assigned to your job. This argument is necessary. See Section 1.9.1 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

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 on a KL10 processor 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 x,...

SET BREAK AT n,,addr ON condition AFTER x,...

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.

x is the number of times the sequence should run before the break occurs. The variable is any decimal number from 1 to 511.

n is a section number ranging from 0-37 octal. The commas separating n and addr are optional.

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:

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).
MUUO	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 MUUO,ALL or MUUO,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
```

When a break occurs, one of the messages:

```
%ADDRESS BREAK AT USER PC xxxxxx

%ADDRESS BREAK AT EXEC xxxxxx UUO AT USER xxxxxx
```

will be printed, and your terminal will be left at monitor level. The second message is produced when MUUO 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

Leaves your terminal at monitor level.

Can be used on 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 MUUO,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. The card file must be in your UFD. 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

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 SETUUO with a different specification.

If the SETUUO overrides the command, the specification given in the monitor call remains in effect until a RESET or EXIT monitor call or another SETUUO 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.

SET CPU ONLY CPxn

Changes the CPU specification so that it includes only the specified CPU.

Where: **x** is:

U designating a logical name,

L designating a KL10 processor,

n is a decimal number from 0 to the number of processors in the system.

Characteristics

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 CPU1.

```
.SET CPU CPU1<RET>
.
```

SET DDT BREAKPOINT Command

Function

The SET DDT BREAKPOINT command enables the DDT breakpoint facility. If the breakpoint facility is on and if DDT is loaded with the program, you can type CTRL/D to interrupt program execution and use DDT to examine the state of the program. When the facility is turned off, CTRL/D is passed to the program.

Format

SET DDT BREAKPOINT argument

Where: the word BREAKPOINT may be omitted.

argument may be ON, OFF, or omitted. If the argument is omitted, SET DDT reports the breakpoint status without changing it.

Characteristics

Leaves your terminal at user level.

Does not destroy your core image.

Restriction

Note that CTRL/D is meaningful only to DDT and VMDDT. It will not have this effect on programs loaded with a compiler-specific debugger like FORDDT.

Example

Enable the DDT breakpoint facility.

```
.SET DDT ON<RET>
[Control-D breakpoint facility is turned on]
```

Debug a MACRO program that increments and prints a number, once per second. Use the DDT command $G to start program execution. Then type CTRL/D to stop the program and examine the current location with DDT. Then type $P to continue execution. Use CTRL/Z to exit to monitor level. Note that CTRL/D is shown here, although it does not echo on your terminal.

```
.DEBUG BPT.MAC<RET>
MACRO: BPT
LINK:  Loading
[LNKDEB DDT execution]
DDT
$G
```

```
This program will increment a count, once per second.

1
2
3
4
5
6
7
8
9
10
<CTRL/D>
$0B>>LOOP#+12/  JRST LOOP#  $P
11
12
13
14
<CTRL/D>
$0B>LOOP#+1/   SLEEP  1,  ^Z

.
```

SET DEFAULT BIGBUF Command

Function

The SET DEFAULT BIGBUF command sets the disk buffer size for the job. Disk I/O can be improved by increasing the disk buffer size, but buffers increase the size of the job.

With the SET DEFAULT BIGBUF command, each added block gives you 200 octal words of disk buffer. For example, SET DEFAULT BIGBUF 5 sets the buffer size to 5 blocks (1200 octal words). SET DEFAULT BIGBUF 0 defaults the buffer size to one block. The maximum value for SET DEFAULT BIGBUF is 31. When big buffers are in use, TOPS-10 defaults to a value of four blocks (1000 octal words).

Because the SET DEFAULT BIGBUF command sets the buffer size for the job, the buffer size will remain the same until the user logs out, or changes it by using the command again.

NOTE

In order to use big buffers, the program that is doing disk I/O must be written to utilize the buffer size. Otherwise, this command has no effect on your job except to increase job size.

Format

SET DEFAULT BIGBUF n

Where: **n** is 0 through 31 decimal. This value indicates the number of blocks (200 octal words) per buffer.

Characteristics

Requires LOGIN.

Leaves your terminal at monitor level.

Does not destroy your core image.

Example

Set the disk buffer size to 5 blocks.

```
.SET DEFAULT BIGBUF 5<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 for disk is 6; for all other devices, the default number is 2. 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 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 for disk, and 2 for all other devices.

Characteristics

Requires LOGIN.

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 during the job. 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>. The default protection code set by using this command will be in effect for any of your programs.

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.

Characteristics

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> dd-mmm-yy DSKC:  [27,5434]
```

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> dd-mmm-yy DSKC:  [27,5434]
PROTEC TS1 0 <175> dd-mmm-yy
```

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> dd-mmm-yy DSKC:  [27,5434]
PROTEC TS1 0 <175> dd-mmm-yy
PROTEC TS2 0 <055> dd-mmm-yy
```

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> dd-mmm-yy DSKC:  [27,5434]
PROTEC TS1 <175> dd-mmm-yy
PROTEC TS2 <055> dd-mmm-yy
PROTEC TS3 <175> dd-mmm-yy

.
```

Note the protection codes.

SET DEFER Command

Function

The SET DEFER command causes all output requests (except disk and tape output) generated by programs run by your job to be deferred. These requests will be processed when you log out. Any files that are generated implicitly by a system program (DIRECT, CREF) and automatically output to the line printer, card punch, plotter, or paper tape punch will be held by QUEUE until you log out. When you log out, the files are output to the appropriate devices, or spooled for output, if spooling is set.

Deferred queuing can also be set by logging in with the /DEFER switch. This function is the default for batch jobs.

To prevent all future requests from being deferred, use SET NODEFER.

To queue any deferred requests that are being held for your job, use the appropriate queue command (CPUNCH, PLOT, PRINT, QUEUE, or TPUNCH) followed by two switches: /DEFERRED and /CREATE. To delete the deferred requests without output, use /DEFERRED and /KILL. /DEFERRED releases any requests that are being held for your job, and processes them according to the other switch that you specify.

Format

SET DEFER

Causes all following output requests from programs to be held until you log out.

SET NODEFER

Causes all following output requests to be handled immediately. This is the default function for timesharing jobs.

Characteristics

Does not destroy your job's core image.

Example

The following example illustrates the use of the SET DEFER command.

Set output requests to be deferred (held) until you log out. Also, if the appropriate devices for your job are not yet spooled, use the SET SPOOL command to spool them. (Refer to the SET SPOOL command.)

```
.SET DEFER<RET>

.SET SPOOL ALL<RET>
```

Compile a FORTRAN program, using the /CREF switch to obtain cross-referenced listing files that can be processed by the CREF program.

```
.COMPILE NUMBER.FOR/CREF<RET>

FORTRAN:  NUMBER
MAIN.
```

When compilation is complete, use the CREF command to run CREF. When deferred spooling is not set, the CREF listing is sent directly to the line printer.

```
.CREF<RET>

CREF:  NUMBER
```

After CREF is finished, check your entries in the line printer queue. The files produced by CREF are deferred, so there are no entries in the queue from your job.

```
.PRINT/CHECK<RET>
[THE QUEUES ARE EMPTY]
```

Use the PRINT/DEFERRED/CREATE command to print the CREF listings. The files are released and then spooled.

```
.PRINT/DEFERRED/CREATE<RET>
[PRINTER JOB NUMBER QUEUED, REQUEST #448, LIMIT 6]
```

Check the printer queue again. The CREF listings are queued.

```
.PRINT/CHECK<RET>

PRINTER QUEUE:
JOB NAME   REQ#   LIMIT            USER
--------   ----   -----   ------------------------
 NUMBER     448       6   MARY MAROTTA [27,5434]
THERE IS 1 JOB IN THE QUEUE (NONE IN PROGRESS)
```

Using SET NODEFER, stop having requests deferred.

```
.SET NODEFER<RET>
```

Using the DIRECTORY command, obtain a file that contains a list of the files in the directory. Using the .LST extension ensures that the file will be deleted from your directory after it is printed.

```
.DIRECT FILES.LST=/L<RET>
   TOTAL OF 42 FILES
```

Check the printer queue. Your job now has two printer requests in the queue.

```
.PRINT/CHECK<RET>

PRINTER QUEUE:
JOB NAME   REQ#   LIMIT            USER
--------   ----   -----   --------------------------
 NUMBER     448       6   MARY MAROTTA [27,5434]
 FILES      449       6   MARY MAROTTA [27,5434]
THERE ARE 2 JOBS IN THE QUEUE (NONE IN PROGRESS)

.
```

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 1.9.1 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

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

Leaves your terminal at monitor level.

Example

```
.SET DSKFUL PAUSE<RET>
.
```

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 FORMAT Command

Function

The SET FORMAT command sets the mode of a tape.

Format

SET FORMAT MTxn:mode

Where: **MTxn:** is the physical or logical name of the magnetic tape device for which the mode is to be set. Refer to Section 1.9.1 for a description of device names.

mode is one of the following:

ANSI (for ANSI-ASCII)
BYTE
DUMP (for COREDUMP)
INDUST (for industry compatible)
SIXBIT
SYSTEM (for system default)

Characteristics

Requires LOGIN.

Leaves your terminal at monitor level.

Does not destroy core image.

Example

Set the format of MTA1 to ANSI mode.

```
.ASSIGN MTA1<RET>
MTA1 assigned
.SET FORMAT MTA1:ANSI<RET>
.
```

SET HOST Command

Function

The SET HOST command connects you to another host computer on the same DECnet or ANF-10 network.

Using the SET HOST command, you can remain at one terminal and switch the terminal connection from one system to a different system.

For ANF-10 networks, 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.

Once a terminal has been connected to a particular ANF-10 system through SET HOST, 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.

On DECnet, when you SET HOST to another system and then log off, you do not remain connected, as you would with ANF-10, but would drop back to your original system.

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

1. If you SET HOST to ANF-10 nodes, the SET HOST command:
   - Does not require LOGIN. However, if you are logged in, your job is detached when your terminal is connected to another system.
   - Leaves your terminal at monitor level.
   - Does not destroy your core image.
2. If you SET HOST to DECnet nodes, the SET HOST command:
   - Requires LOGIN
   - Places your terminal at user level.
   - Destroys your core image.
   - Runs CTHNRT.

Restriction

You cannot SET HOST from the TOPS-10 system through DECnet if you used SET HOST to get to the TOPS-10 system over a DECnet link.

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 attempt to SET HOST to an ANF-10 node, identifying the node by number:

```
.SET HOST 26 <RET>

[BC173B KL #1026/1042] hh:mm:ss TTY72 system 1026/1042
Connected to Node KL1026(26) Line # 0

Please LOGIN or ATTACH

.
```

2. You attempt to SET HOST to an ANF-10 node, identifying the node by name:

```
.SET HOST TWINKY<RET>

[Twinky KL702/21A] dd-mmm-yy hh:mm:ss TTY111 System 2197
Connected to Host TWINKY(77) via node KL1026(26) Line # 55

Please LOGIN or ATTACH

.
```

3. You SET HOST to a DECnet node, log in, and log out again. Notice that after logging out you are returned automatically to your original node.

```
.SET HOST QARRY<RET>

[Connected to VMS system QARRY::, using CTERM protocol]
[Type ^\,<RET> to return]

Username: FRANCINI<RET>
Password: <RET>

        This is MicroVMS Version V_._ on node QARRY

    Last interactive login on Thursday, DD-MMM-YYYY 18:04
    Last non-interactive login on Wednesday, DD-MMM-YYYY 18:29
$
$ LOGOUT<RET>
  FRANCINI     logged out at DD-MMM-YYYY 15:31:07.28
[Connection closed by remote:  User unbind request]
.
```

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

Does not destroy your core image.

Leaves your terminal at monitor level.

Requires LOGIN.

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.

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 or nK
             GUIDELINE  nP or 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

Requires LOGIN.

Leaves your terminal at monitor level.

Does not destroy your core image.

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 controls the function of the DX10 tape drive controller. When retry is set, the controller recovers from soft errors, and it reports both soft and hard errors in SYS:ERROR.SYS. When retry is off, the controller does not recover from soft errors and all errors are reported as hard errors in ERROR.SYS.

Format

SET RETRY MTxn: ON

This format of the SET RETRY command sets the recovery function. This is the default function of the controller.

SET RETRY MTxn: OFF

This format of the command turns off the recovery function.

Where: **MTxn:** is the individual device name of the magnetic tape drive. If you have assigned a logical name to the drive, you may substitute the logical name for the device name.

Characteristics

Leaves your job at monitor level.

Does not affect your job's core image.

Example

The following example shows how the recovery function is set:

```
.SET RETRY MTA5: ON<RET>
.
```

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 1.9.1 for device name formats.

Characteristics

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].

Examples

1. Add the card punch to your list of spooled devices.

   ```
   .SET SPOOL CDP:<RET>
   .
   ```

2. Delete the line printer from your list of spooled devices.

   ```
   .SET SPOOL NO LPT:<RET>
   .
   ```

3. Clear your list of spooled devices.

   ```
   .SET SPOOL NONE<RET>
   .
   ```

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

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
```

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 2-1.

Table 2-1: Terminal Hardware Characteristics

TYPE	LOWERCASE	TABS	FORM	WIDTH	DISPLAY	FILL	LENGTH	ALTMODE
GIGI	Y	Y	N	80	Y	0	24	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
LA120	N	Y	Y	132	N	0	N/A	N
LA180	N	Y	Y	132	N	0	N/A	N
LN01S	Y	Y	Y	80	N	0	66	N
LN03	Y	Y	Y	80	N	0	66	N
LT33	N	N	N	72	N	1	N/A	Y
LT35	N	Y	Y	72	N	1	N/A	Y
TTY	N	N	N	72	N	1	N/A	Y
TTY33	N	N	N	72	N	1	N/A	Y
TTY35	N	Y	Y	72	N	1	N/A	Y
VK100	Y	Y	N	80	Y	0	24	N
VT05	N	Y	N	72	Y	2	20	N

Table 2-1: Terminal Hardware Characteristics (Cont.)

TYPE	LOWERCASE	TABS	FORM	WIDTH	DISPLAY	FILL	LENGTH	ALTMODE
VT06	N	N	N	72	Y	1	25	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
VT100	Y	Y	N	80	Y	0	24	N
VT101	Y	Y	N	80	Y	0	24	N
VT102	Y	Y	N	80	Y	0	24	N
VT103	Y	Y	N	80	Y	0	24	N
VT125	Y	Y	N	80	Y	0	24	N
VT131	Y	Y	N	80	Y	0	24	N
VT180	Y	Y	N	80	Y	0	24	N
VT185	Y	Y	N	80	Y	0	24	N
VT200	Y	Y	N	80	Y	0	24	N
VT220	Y	Y	N	80	Y	0	24	N
VT240	Y	Y	N	80	Y	0	24	N
VT300	Y	Y	N	80	Y	0	24	N
VT320	Y	Y	N	80	Y	0	24	N
VT330	Y	Y	N	80	Y	0	24	N
VT340	Y	Y	N	80	Y	0	24	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 LENGTH for hard-copy terminals.

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 (right brace: }) and 176 (tilde: ~).
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 VT52 Owner's Manual.
CRLF	Controls the automatic carriage return and line feed at the end of the line. Used with the WIDTH argument, CRLF gives an automatic carriage return/line feed at the edge of the terminal screen. NO CRLF will cause lines wider than the set WIDTH to be truncated. The default setting varies according to the terminal type.
DEBREAK	No longer supported.
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 hold the characters until the program requests input; NO DEFER allows them to be echoed on your terminal as soon as normal output is complete.
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 on the terminal. NO ECHO suppresses echoing of all characters you type.
EIGHTBIT	Enables the terminal to accept and generate eight bit characters. A terminal set with the EIGHTBIT switch will be able to communicate with a program that uses seven bit characters.

ELEMENT nnn	No longer supported.
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). With the DN87S front-end, there will be a short time lapse before each character is sent. Table 2-2 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.

Table 2-2: 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 (1)
LF	012	0	2	6	6
VT	013	0	2	6	6
FF	014	0	12	21	21
CR on output	015	0	1	3	3 (2)
automatic CR (3)		0	2	4	4
CRLF on output	015-012	0	3	9	9 (4)
XON	021	1	1	1	1
TAPE	022	1	1	1	1
XOFF	023	1	1	1	1
NTAP	024	1	1	1	1

(1) 1 if 0-3 spaces to tab stop; 2 if 4-7 spaces to tab stop.
(2) Output only; no fillers on input.
(3) Refer to the SET TTY CRLF command.
(4) Sum of the fillers output for a CR and LF.

Argument	Function
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.
IGNORE	Sets input speed to 0, making terminal useless.
ISO	Controls compatibility with ISO (International Standards Organization) Latin-1 supplemental graphic character set. NO ISO means DEC Multinational Character Set. To display the current setting, you must use the command INITIA ATTRIBUTES.
LC	Controls the translation of lowercase letters to uppercase letters. NO LC allows all characters to be translated to uppercase by the monitor. 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.
LENGTH n	Defines the number of lines of your terminal page. To stop output on your terminal after a page of length n, use the arguments STOP or PAGE. SBELL, SSTOP, STOP, XONXOF will use this value as the page length.
LOCALCOPY	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. NO LOCALCOPY is the default setting. Use LOCALCOPY for a local-copy terminal.

OVERSTRIKE	Indicates the terminal allows a three-character sequence to create one composite output character by printing one character, backspacing, and then printing another character over the first. To display the current setting of this attribute, you must use the command INITIA ATTRIBUTES.
PAGE n	Defines the number of lines on your terminal page, enables recognition of CTRL/Q and CTRL/S (XON/XOFF; see Section 1.6.) and stops output to your terminal after n lines. That is, TTY PAGE n is the same as TTY LENGTH n XONXOF STOP. The default value of n is determined by the declared terminal type. TTY PAGE and TTY PAGE 0 have the same effect as TTY XONXOF. 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.
QUOTE	Causes a ^V (control-V) and character to behave as a single super-quote character. They will not echo separately, and a single delete erases them both. NO QUOTE is the default argument.
REMOTE	Clears the "local" access characteristic for your job. REMOTE is not recommended for use by non-privileged users because the action can be reversed only by an operator [1,2]. The monitor does not allow CTY: to be set REMOTE. NO REMOTE sets your terminal so that you can log into an account that requires local access. For specific information on account characteristics, including local and remote access types, refer to the description of REACT in the TOPS-10 Software Installation Guide.
RTCOMPATIBILITY	Disables the CTRL/R and CTRL/T features. (Refer to Section 1.6 for a description of CTRL/R and CTRL/T.) NO RTCOMPATIBILITY is the default argument.
SBELL	Sets the terminal to ring the bell when output is automatically stopped (this is the default action). NO SBELL suppresses the bell.

SLAVE	Specifies that the terminal becomes slaved, so that no commands can be typed on the terminal. This is useful for terminals that are ASSIGNed by another job. This command is not recommended for use by non-privileged users because the actions can be reversed only by an operator [1,2], using NO SLAVE.
SPEED m n	Sets the terminal transmitting speed to m and the receiving 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. NO SPEED is an invalid argument. NOTE Certain hardware configurations do not allow transmit and receive speeds to be different (also called "split speed").
SSTOP n	Ignores CTRL/Q until output is stopped either automatically by the system, or by CTRL/S from user. The page length n is optional.
STOP n	Stops output to the terminal automatically after n lines. The page length n is optional. When you type CTRL/Q, the page length counter is reset to 0.
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	Causes CTRL/S and CTRL/Q to stop and start the paper tape reader on a Teletype. Refer to the TOPS-10 Monitor Calls Manual. NO TAPE, the default, causes CTRL/S and CTRL/Q to stop and resume output from the system to the terminal. Refer to Sections 1.6.7 and 1.6.8.
TIDY	No longer supported.

TYPE nnnn

Sets terminal characteristics defined by the declared terminal type (nnnn).

For a list of the terminal types recognized by the system, type:

HELP *

The output from HELP * includes a list of supported terminal types.

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.

The characteristics that are set by declaring the terminal type are:

LOWERCASE
TABS
FORM
WIDTH
DISPLAY
FILL
PAGE
ALTMODE

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.

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.

XONXOF

Allows you to use CTRL/S (XOFF) and CTRL/Q (XON) to stop and resume system output to the terminal. This is the default. NO XONXOF prevents the system from recognizing CTRL/S and CTRL/Q.

Characteristics

Leaves your terminal at monitor level.

Does not require LOGIN.

Restrictions

The SET TTY (or TTY) command is not available to batch jobs.

Associated Commands

INITIA displays the TTY parameters

Example

Tell the system that your terminal is a VT340.

```
.SET TTY VT340 <RET>
```

Issue the INITIA TTY command. The system responds with the terminal characteristics of a VT340 terminal.

```
.I TTY <RET>

RL353A DEC10 Development 16:02:49 TTY162 system 1026/1042/1322
Connected to Node KL1026(26) Line # 162

TYPE:VT340   MODEL:VT340     CLASS:VT300     APC:HARDWIRED
ECHO         WIDTH:80        LENGTH:24       NOSTOP
FILL:0       LC              TABS            NOFORM
CRLF         NOGAG           SBELL           DISPLA
NOTAPE       BLANKS          NOALTMOD        NORTCOMP
NOREMOTE     XONXOF          NOUNPAUS        NOESCAPE
NOEIGHTB     NOQUOTE         IDLEDI:0        EDITOR
NOSLAVE

.
```

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

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>
```

```
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 provides a means with which to measure the performance status of your program. This command tells the system to print (or suppress) job statistics.

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:

DAY	WRITE
VERSION	MTA
READ	CONTEXTS
RUN	FILES
WAIT	

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, GETSEG, or SEGOP. monitor call; or if a DAEMON wait occurs.

The following arguments print a message 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.

When an UNLOAD command, the unload function of a UUO, or a DEASSIGN command is executed,

MTA Prints magnetic tape statistics in the form:

[MTxn:reelid READ (C/H/S) = a/b/c WRITE (C/H/S) = d/e/f]

Where: **x** represents the tape controller

n represents the drive unit 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,...], error z]

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 fail. Error codes can be found in the TOPS-10 Monitor Calls Manual.

Note that, in the case of execute-only jobs, this message is suppressed.

Whenever the current context changes, (by the use of the CONTEXT, PUSH, or POP commands),

CONTEXTS Prints information about the current context. The message is in the form:

[CONTEXT contextname (n) prog]

Where: **contextname** is the name (if any) of the current context

N is the current context number

prog is the name of the program (if any) that the current context has stored in core.

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.

Associated Messages

If you type SET WATCH with no arguments, or with an invalid argument, the following error message occurs:

?ARGS ARE: CONTEXTS,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

Leaves your terminal at monitor level.

Example

List the arguments:

```
.SET WATCH<RET>
?ARGS ARE: CONTEXTS,DAY,RUN,WAIT,READ,WRITE,VERSION,MTA,
FILES,ALL,NONE
```

Set the statistics for time, run time, wait time, version number:

```
.SET WATCH DAY<RET>

.SET WATCH RUN<RET>

.SET WATCH WAIT<RET>

.SET WATCH VERSION<RET>
```

Run a program:

```
.RUN LOOP<RET>
[hh:mm:ss]
?LOOP.SAV NOT FOUND
[0.06  0.91    ]

.R PIP<RET>
[hh:mm:ss]

[S:PIP  33B(260) + ]
*C
[0.08 6.08    ]

.R TECO<RET>
[hh:mm:ss]

[S: TECO 230(162) + ]
*C
[2.05 6.10    ]

.
```

SHOW ALLOCATION Command

Function

The SHOW ALLOCATION command prints information on your terminal about the resources that are allocated to your job. SHOW ALLOCATION prints the names of the tape and disk volumes and volume sets that are allocated and mounted for your job, and the drives on which they are mounted.

The information is printed on your terminal in the following form:

ALLOCATION FOR JOB x user-name [ppn]

VOLUME SET	RESOURCES	TYPE	ALL	OWN
name	volume	description	no.	no.

Where:

x is the number of your job.

user-name is your user name.

[ppn] is your project-programmer number.

name is the name of the volume set, if it applies. For example, if the resource is a drive, "---" is printed in this column.

volume is the name of the volume that is allocated or mounted for your job. If the resource is a tape volume set, the name shows the volume that is currently mounted. Disk structure names, disk drives, and tape drives are also shown.

description is the description of the type of volume. This identifies whether the resource is a structure, disk drive, magtape drive, or magtape volume. The description is useful when resources have identical names.

no. is the number of disk packs, drives, or tapes that are allocated to your job and that are mounted for your job for each resource.

Format

SHOW ALLOCATION

Characteristics

Runs the QUEUE program.

Requires LOGIN.

Destroys your core image.

Example

The following example shows the use of the ALLOCATE, DEALLOCATE, MOUNT, DISMOUNT, and SHOW ALLOCATION commands. The resources are reserved for a multivolume tape volume set with the ALLOCATE command. The name of the volume set is TAPE-SET, and it contains three volumes. The logical name TS is assigned to the tape set. The tape is write enabled, and it does not have standard labels.

```
.ALLOCATE TAPE-SET(VOL1,VOL2,VOL3):TS/WRITE-ENABLE/LABEL:NONE<RET>
[ALLOCATE REQUEST TS QUEUED, REQUEST #672]
```

A file structure named DSKR: is mounted for the job:

```
.MOUNT DSKR:<RET>
[MOUNT REQUEST DSKR QUEUED, REQUEST #673]
[STRUCTURE DSKR MOUNTED]
```

The job's resources are shown using the SHOW ALLOCATION command:

```
ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE         ALL  OWN
---------------------- ----------------  ------------  ---  ---
---                    9 TK 800/1600     MAGTAPE UNIT   1    0
---                    RP06              DISK UNIT      2    2
---                    RP20              DISK UNIT      1    1
DSKB                   DSKB              STRUCTURE      1    1
DSKC                   DSKC              STRUCTURE      1    1
DSKR                   DSKR              STRUCTURE      1    1
TAPE-SET               VOL1              MAGTAPE VOL.   1    0
TAPE-SET               VOL2              MAGTAPE VOL.   1    0
TAPE-SET               VOL3              MAGTAPE VOL.   1    0
```

The tape set is mounted, and the resources are again displayed:

```
.MOUNT TS<RET>
[MOUNT REQUEST TS QUEUED, REQUEST #673]
[MAGTAPE TS MOUNTED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE         ALL  OWN
---------------------- ----------------  ------------  ---  ---
---                    9TK 800/1600      MAGTAPE UNIT   1    1
---                    RP06              DISK UNIT      2    2
---                    RP20              DISK UNIT      1    1
DSKB                   DSKB              STRUCTURE      1    1
DSKC                   DSKC              STRUCTURE      1    1
DSKR                   DSKR              STRUCTURE      1    1
TAPE-SET               VOL1              MAGTAPE VOL.   1    1
TAPE-SET               VOL2              MAGTAPE VOL.   1    0
TAPE-SET               VOL3              MAGTAPE VOL.   1    0
```

After work is finished by accessing the tape set and the structure, the structure is dismounted. Because the structure was no explicitly allocated, it is automatically deallocated.

```
.DISMOUNT DSKR<RET>
[VOLUME SET DSKR HAS BEEN DISMOUNTED]
```

The tape volume set is dismounted:

```
.DISMOUNT TS<RET>
[VOLUME SET TS DISMOUNTED]
```

The job's resources are displayed:

```
.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE         ALL  OWN
---------------------- ----------------- ------------  ---  ---
---                    9 TK 800/1600     MAGTAPE UNIT   1    0
---                    RP06              DISK UNIT      1    1
---                    RP20              DISK UNIT      1    1
DSKB                   DSKB              STRUCTURE      1    1
DSKC                   DSKC              STRUCTURE      1    1
TAPE-SET               VOL1              MAGTAPE VOL.   1    0
TAPE-SET               VOL2              MAGTAPE VOL.   1    0
TAPE-SET               VOL3              MAGTAPE VOL.   1    0
```

At this point, the tape set can again be mounted, or it can be deallocated. The tape set is deallocated:

```
.DEALLOCATE TS<RET>
[VOLUME SET TS HAS BEEN DEALLOCATED]

.SHOW ALLOCATION<RET>

ALLOCATION FOR JOB 59 MARY MAROTTA [27,5434]
     VOLUME SET           RESOURCES          TYPE         ALL  OWN
---------------------- ----------------- ------------  ---  ---
---                    RP06              DISK UNIT      1    1
---                    RP20              DISK UNIT      1    1
DSKB                   DSKB              STRUCTURE      1    1
DSKC                   DSKC              STRUCTURE      1    1
```

SHOW QUEUES Command

Function

The SHOW QUEUES command displays a list of the entries in the system queues on your terminal. You can specify the system queues that you want displayed, or you can see all of the system queues. SHOW QUEUES lists the mount queue, batch queue, event queue, and output queues.

Format

SHOW QUEUES argument/switches

Where:

argument specifies the queue or queues to be displayed. The argument is optional. If you do not specify an argument, all of the system queues are displayed.

The following arguments are valid:

ALL-REQUESTS	Shows queue requests in all system queues. This is the default function.
BATCH-REQUESTS	Shows batch requests.
CARD-PUNCH-REQUESTS	Shows card punch requests.
EVENTS-REQUESTS	Shows scheduled system events.
MOUNT-REQUESTS	Shows mount requests.
OUTPUT-REQUESTS	Shows requests in all output queues.
PAPER-TAPE-REQUESTS	Shows paper tape punch requests.
PLOTTER-REQUESTS	Shows plotter requests.
PRINTER-REQUESTS	Shows line printer requests.

/switches is one or more of the following switches. The switches define the amount of information to be displayed about each queue request and which queue requests to display. By default, SHOW QUEUE displays all the system queues for all jobs.

Switch	Function
/ALL	Displays a list of all the queue requests. This is the default function. When showing the mount requests, /ALL prints a list of all the structures that are mounted and, for each structure, a list of every job that has the structure in its search list.
/BRIEF	Displays a short description of each request.
/FULL	Displays all available information about each request.
/HELP	Prints information about SHOW ALLOCATION and SHOW QUEUES. To use this switch, type: SHOW/HELP.
/USER:[ppn]	Displays the queue requests for the specified user. The project-programmer number must be typed in brackets.

Associated Messages

[THE QUEUES ARE EMPTY]

This message occurs when there are no queue requests that match the specification.

Characteristics

Runs the QUEUE program.

Destroys your core image.

Leaves your job at monitor level.

Example

1. The following example illustrates the output from the SHOW QUEUES command, used with the MOUNT-REQUESTS argument:

```
.SHOW QUEUES MOUNT-REQUESTS<RET>

MOUNT QUEUE:
VOLUME  STATUS  TYPE  WRITE  REQ#   JOB#      USER
------  ------  ----  -----  ----   ----   ------------
MTB2    WAITING TAPE  LOCKED  614     52    MAROTTA [27,5434]
  VOLUME-SET: MTB2
  LABEL-TYPE: BYPASS, TRACKS: 9, DENSITY: 1600 BPI
THERE IS 1 REQUEST IN THE QUEUE
```

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 MTxn: x FILES

Advances forward x files.

SKIP MTxn: x RECORDS

Advances forward x records.

SKIP MTxn: EOT

Advances forward to the logical end of tape.

Where: **MTxn:** is the physical device name of the magnetic tape unit. Device names are discussed in Section 1.9.1.

x is a decimal number representing the number of records or files to skip over.

Characteristics

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

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
```

SAVE the program.

```
.SAVE PROG<RET>
PROG SAVED
```

START the program.

```
.START<RET>

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. Refer to the QUEUE command for further information and examples.

Refer to the TOPS-10/TOPS-20 Batch Reference Manual for more information about batch jobs.

Format

SUBMIT jobname=control-file-spec,log-file-spec

Where: **jobname** is the name of the job to be entered into the queue. The jobname is optional. If you omit it, the default job name is the name of the log file.

The equal sign is required only if you specify the job name.

control-file-spec is the name of the input file. This file contains all monitor-level and user-level commands for processing by the batch controller (BATCON). If you do not include a device in the file specification, and if the monitor does not find the control file in your job search list, the monitor will search ersatz device CTL for the control file.

log-file-spec is the name of the output file. The batch controller uses this file to record its processing of the job.

Only the two files mentioned above can be specified in a request to the batch input queue. The name of the control file is required; the log file name is optional and, if omitted, is taken from the control file. If the job name is omitted, it defaults to the name of the log file, if present, or the name of the control file if no log file is specified. If an extension is omitted, the following are assumed:

.CTL for the control file
.LOG for the log file

If you type SUBMIT with no arguments or switches, a list of the jobs in the batch queue will be printed on your terminal.

The switches to this command can be divided into two categories, depending on whether the switch can be used only once, or can be used more times, in a single command string. The two categories are:

- **Queue-Operation Switches**

  These switches can be used only once in the command string. They affect the entire request, and you can place them anywhere in the command string. If you have used one of these switches in a command string, you cannot use it again in the same string. Many switches have a /NO construction, which has a negative effect. Be sure you do not use the /NO construction of a switch in the same command string with the positive construction.

- **File-Control Switches**

  These switches can be used any number of times in the command string. You can also use the /NO construction of a switch in the same command string with the positive construction. To achieve a temporary or permanent effect by the placement of the switch, refer to Section 1.8.4.

Switch	Category	Function
/ABEFORE: date-time	File control	Queues the file only if the access date is before the specified date and time.
/ACCOUNT: "string"	Queue operation	Specifies the account to which the job should be charged. If the account contains any nonalphanumeric characters, you must enclose the string in quotation marks.
/AFTER: date-time	Queue operation	Processes the request after the specified date and time.
/ALLFILES: YES or NO	Queue operation	Accepts the request only if all of the files in the request exist. That is, if any specified file is not present, no files are processed. /ALLFILES is the same as /ALLFILES:YES. By default (or /ALLFILES:NO), if all of the files do not exist, the existing files are still processed.
/ASINCE: date-time	File control	Queues only the files that have been accessed since the specified date and time.
/ASSISTANCE: YES or NO	Queue operation	Specifies whether the job needs or does not need operator intervention. Arguments are YES or 1 and NO or 0. If you specify NO and then request assistance, your job is cancelled. Assistance is any action the operator must take before the job can continue, including PLEASE and MOUNT requests.
/BATLOG:arg	Queue operation	Controls the output of the log file on disk. Arguments are: APPEND Appends the log file to any existing file of the same name. SUPERSEDE Replaces any existing file of the same name with the new file.

		SPOOL Spools the log file for output to the printer instead of writing the file into your directory. This prevents a large log file from using up your disk quota. Will not override /OUTPUT:NOLOG.
/BATOPT: option-name	Queue operation	Specifies a LOGIN option line to read for LOGIN switches to apply to the batch job. The option name that you specify with the /BATOPT switch must match a line in the SWITCH.INI file that appears as: LOGIN:option-name/switches
/BEFORE: date-time	File control	Queues the file only if it was created before the specified date and time.
/BEGIN:n	File control	Starts the output on the nth line of the control file.
/CARDS:n	Queue operation	Uses n as the maximum number of cards that can be punched by the batch job (up to 10,000). If you omit the switch, no cards will be punched.
/CHECK	Queue operation	Prints on your terminal a list of the batch queue entries made by your job.
/CORE:n	Queue operation	Uses n (in decimal K) as the maximum amount of memory that your job can use.
/CREATE	Queue operation	Makes a new entry in the batch input queue. This is the default function except when you are listing queue entries.
/DELETE	File control	Deletes the control file after processing it. (This is the same as /DISPOSE:DELETE.)
/DEPENDENCY:n	Queue operation	Specifies the initial value of the dependency count in decimal. When used with /MODIFY, this switch changes the dependency count of a previously submitted job. If n is a signed number (+ or -), that number is added to or subtracted from the dependent job's count. If n is not a signed number, the dependent job's count is changed to n.

/DESTINATION: node	Queue operation	Specifies the node that will process output requests from this batch job, including the printing of the log file. Use the node name or the node number to specify the node. If you specify another host or a node that does not exist, the job will wait in the queue indefinitely.
/DISPOSE: arg	File control	Controls the disposition of the control file after it is processed. The arguments to this switch are: DELETE deletes the file from your directory after spooling it. PRESERVE preserves the file after processing it. This is the default function. RENAME is no longer supported.
/DISTRIBUTION: "text"	Queue operation	Specifies text to place in the distribution field, on the banner page of output listings. For batch input requests, the distribution text is printed on the banner page of the log file listing. You can use this field to include mailing information, or the location where the operator should leave the listing. The text field may be up to 39 alphanumeric characters, including punctuation and spaces if the text is placed in quotation marks.
/ERBINARY	File control	Prints an error message if a binary file is included in the request. This is the default function.
/ERNONE	Queue operation	Prints an error message if no files match the file specification. This is the default function.
/ERPROTECTION	Queue operation	Prints an error message if processing the request would require a violation of file protection. This is the default function.
/FAST	Queue operation	Prints a list of the entries in the batch queue in a fast format on your terminal.
/FEET:n	Queue operation	Uses n as the maximum number of feet of paper tape that the batch job can punch. If the switch is omitted, no paper tape is punched.

/HELP: arg	Queue operation	Prints information on your terminal about the QUEUE command. This switch does not queue any files. /HELP can be used alone or with one of the following arguments: TEXT prints a message about the format and switches to the QUEUE command. This is the same as /HELP with no arguments. SWITCHES prints a list of all the switches available with the QUEUE command.
/JOBNAME:name	Queue operation	Specifies the name of the job. The job name can be up to six alphanumeric characters.
/KILL	Queue operation	Removes the specified entry from the queue. You must give the job name, /SEQUENCE, or /REQUESTID, to the left of the equal sign in the command line. Refer to the examples.
/LENGTH:n:m	File control	Processes only files whose length is between n and m blocks.
/LIST:arg	Queue operation	Prints information about the jobs in the queue. If you use /LIST alone, it shows the jobs in the queue. This is equivalent to using the QUEUE command with no arguments and no switches. /LIST can be abbreviated to /L. The switch can also take one of the following arguments: ALL shows all data about each queue request. FAST shows a fast list of the queue requests. (This is the same as /FAST.) JOBS shows a list of the jobs in the queue. (This is the same as /LIST with no arguments.) SUMMARY shows only the summary line of the queue display.
/MESSAGE:arg	Queue operation	Specifies the amount of information to be printed when an error occurs from the request. You can specify one or more of the following arguments: ADDRESS prints the location in memory where the error occurred. CONTINUATION prints information about the error.

		FIRST prints the one-line error message. PREFIX prints a six-character error prefix.
/METERS:n	Queue operation	Uses n as the maximum number of meters of paper tape that can be punched by the job.
/MODIFY	Queue operation	Alters the specified parameter in the specified job. This switch requires that you have access rights to the job. You must give a job name, /SEQUENCE, or /REQUESTID, to the left of the equal sign in the command line. This switch can be used to modify a previously submitted request as long as the request has not been started. Refer to the examples.
/NEW: YES or NO	File control	Accepts the request even if the file does not yet exist. Does not search for control file on CTL:
/NONEW	File control	Does not accept the file specification of a file that does not exist. This is the default function.
/NONOTIFY	Queue operation	Does not set the system to notify you when a request is completed. Refer to /NOTIFY.
/NOOPTION	Queue operation	Ignores the SWITCH.INI file. SWITCH.INI files are described in Appendix B.
/NOPHYSICAL	File control	Searches for the file by recognizing any logical names. This is the default function.
/NORESTART	Queue operation	Prevents the job from being restarted if it was stopped because of a system crash. This is the default function.
/NOTIFY: YES or NO	Queue operation	Sets the system to notify you when your request is completed. To be notified, use /NOTIFY with no argument, or with YES or 1 as an argument. To suppress notification, use /NOTIFY:0 or /NOTIFY:NO. By default, you are not notified when a request is finished.
/OKBINARY	File control	Accepts files whose extensions indicate that they include binary information. Normally, files with extensions .SAV, .SHR, .LOW, .REL, .EXE, and .HGH will not be accepted for processing.

/OKNONE	Queue operation	Does not produce a warning message if no files match the file specification.
/OKPROTECTION	Queue operation	Does not output an error message when a protection code is violated.
/OPTION:option	Queue operation	Uses the option line QUEUE:option in the SWITCH.INI file. SWITCH.INI files are discussed in Appendix B.
/OUTPUT:arg	Queue operation	Determines whether or not the log file will be printed. This switch has three arguments only one of which can be specified at a time. 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.
/PAGES:n	Queue operation	Uses n as the maximum number of pages of output that your job can print.
/PATH:[dir]	Queue operation	Specifies the directory to be accessed.
/PHYSICAL	File control	Does not recognize logical names for devices in the command line.
/PRESERVE	File control	Saves the control file after processing it. This is the default and it is the same as /DISPOSE:PRESERVE.
/PRIORITY:n	Queue operation	Gives the specified priority (n is 1 to 63) to the request. A larger number has greater priority.
/PROCESSING:node	Queue operation	Specifies the node that is to process the batch job. Use the node name or node number to specify the node. Batch jobs can be submitted to IBM host nodes only. Jobs submitted to nodes other than IBM host nodes will wait in the queue indefinitely.
/PROTECTION: nnn	Queue operation	Specifies a protection code for the queue request. Queue requests are protected in the same way that files are protected. Refer to Section 1.9.4.
/READER	Queue operation	Causes a disk-resident card job to be read as if it were punched on cards and had been submitted through the card reader.

/REMOTE	Queue operation	Prints on your terminal a list of remote queues. Must be used with /DESTINATION.
/REQUESTID:n	Queue operation	Specifies the request identification number of the job you wish to modify or terminate (/KILL). The request identification number is assigned when the request is queued. This switch is used to the left of the equal sign in the command line. Refer to the examples.
/RESTARTABLE: YES or NO	Queue operation	Specifies whether the job should be restarted after the system has crashed and been restored. Arguments are: YES (or 1) and NO (or 0). The default function is NO.
/RUN:file	Queue operation	Executes the specified program (file) after your request is accepted.
/RUNCORE:nx	Queue operation	Executes the program specified in /RUN in nK of core after the request is accepted. The value can also be expressed in terms of nP (pages).
/RUNOFFSET:n	Queue operation	Executes the program specified in /RUN with offset n after the request is accepted.
/SEQUENCE:n	Queue operation	Specifies a sequence number to help identify a request to be modified or deleted. This switch must be used to the left of the equal sign in the command line.
/SINCE: date-time	File control	Queues only the files with creation dates after the specified date and time.
/SITGO	Queue operation	Processes the batch job using the SITGO compiler.
/STREAM:n	Queue operation	Prints a list of the jobs that are running or destined to run in the specified batch stream.
/TAG:xxx	File control	Starts at the statement labeled xxx in the control file. Equivalent to GOTO xxx at the beginning of the control file.
/TIME:hh-mm-ss	Queue operation	Specifies the CPU time limit for the job. The form /TIME:n can be used to specify a limit of n seconds.

/TMPFIL: file:text	Queue operation	Creates a temporary file TMP:file and enters the text into this file.
/TPLOT:n	Queue operation	Uses n minutes as the maximum amount of plotting time allowed for your job. If you omit the switch, no plotter time is allowed.
/UNIQUE: YES or NO	Queue operation	Specifies whether more than one batch job can run from your PPN at one time. If the value is YES (or 1), only one job will run at a time. Any other batch jobs will wait until the previous job is finished. If the value is NO (or 0), any number of batch jobs can run at the same time.
/USERNAME: "name"	Queue operation	Specifies the user name field for the banner page of output listings. For batch input requests, the user name is printed on the banner page for the log file listing. This field can contain up to 39 alphanumeric characters, and may include punctuation and spaces if the name is placed in quotation marks.

Associated Messages

When you make an entry into the batch queue, the system prints the following message on your terminal:

[BATCH JOB name QUEUED, REQUEST #nnn, RUN TIME:time]

Where: **name** is the name of the job in the queue. This can be specified by the user. Otherwise, the name of the log file is used.

nnn is the number that represents the request identification of the job in the queue.

time is the amount of time allowed for the batch job to run.

Characteristics

Leaves your job at monitor level.

Destroys your core image.

Does not require that you be logged in if you only want the list of batch queue entries.

Examples

1. Create a control file.

```
.SOS CUE.CTL<RET>

INPUT:  CUE.CTL
00100 PRINT SYS:  NOTICE.TXT<RET>
00200 <ESC>
E<RET>

[DKSC:  CUE.CTL]
```

Submit the control file.

```
.SUBMIT CUE.CTL<RET>

[BATCH JOB CUE QUEUED, REQUEST #132, LIMIT 0:05:00]
```

Use DIRECTORY to see new .LOG file.

```
.DIR CUE<RET>

CUE CTL 1 <055> dd-mmm-yy DSKC:  [27,5434]
CUE LOG 3 <055> dd-mmm-yy
TOTAL OF 4 BLOCKS IN 2 FILES ON DSKC:  [27,5434]

.
```

2. Submit the control file CUE.CTL to the batch queue for processing at 5:00 p.m.:

```
.SUBMIT CUE /AFTER:17:00<RET>
[BATCH JOB CUE QUEUED, REQUEST  79, LIMIT 0:05:00]
```

Change the processing time to 4:30 p.m., identifying the job by its request identification number:

```
.SUBMIT /REQUESTID:79= /MODIFY /AFTER:16:30<RET>
[1 JOB MODIFIED]
```

Cancel the batch job CUE altogether:

```
.SUBMIT CUE= /KILL<RET>
[1 JOB CANCELED]

.
```

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, system uptime, CPU uptime on an SMP system, 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.

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.

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

Leaves your terminal at monitor level.

Destroys your core image.

Does not require LOGIN.

Example

If you wanted to use the ASSIGN command to assign LPT 260 to your job, you could use the B argument to SYSTAT to see if the printer is busy.

```
.SYSTAT B<RET>

Busy devices:
Device  Job     Why     Logical

TTY52    1      init
TTY2    21      init
DET60    1      init
MPX1    21      init
LPT260  14      init
TSK26   26      init
TSK26   26      init
TSK26   26      init
TSK26   18      init
```

SYSTAT B lists LPT260 as a busy device, therefore, you cannot assign it to your job.

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]

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.

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

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>
```

.

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

Leaves your terminal at monitor level.

Does not require LOGIN when requesting time for another user's job.

Does not destroy core image.

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 places entries into the paper tape punch output queue. Refer to the QUEUE command for further information and examples.

Format

```
TPUNCH dev:jobname=file-spec
```

Where:

dev: is the name of the specific tape punch on which your files should be processed. (For example, PTP2: is tape punch number 2.) You can have your files processed on another node by using the format PTPSxx:, where xx is the node number. (For example, PTPS25: is a punch on node 25.) The device name is optional.

jobname is the name of the job to be entered into the queue. The default is the name of the first file in the request.

The equal sign is required if you specify the job name, the device name, or both.

file-spec is a single file specification or a string of file specifications, separated by commas, for the files to be processed. A file specification is in the form dev:file.ext[directory].

If you do not enter a job name or an input specification, the system prints a list of the jobs in the paper tape punch queue on your terminal.

The switches to this command can be divided into two categories, depending on whether the switch can be used only once, or can be used more times, in a single command string. The two categories are:

- **Queue-Operation Switches**

  These switches can be used only once in the command string. They affect the entire request, and you can place them anywhere in the command string. If you have used one of these switches in a command string, you cannot use it again in the same string. Many commands have a /NO construction, which takes a negative effect. Be sure you do not use the /NO construction of a switch in the same command string.

- **File-Control Switches**

  These switches can be used any number of times in the command string. You can also use the /NO construction of a switch in the same command string. To achieve a temporary or permanent effect by the placement of the switch, refer to Section 1.8.4.

Switch	Category	Function
/ABEFORE: date-time	File control	Queues the file only if the access date is before the specified date and time.
/ACCOUNT: "string"	Queue operation	Specifies the account to which the job should be charged. If the account string contains any nonalphanumeric characters, you must enclose the string in quotation marks.
/AFTER: date-time	Queue operation	Processes the request after the specified date and time. Refer to Section 1.8.3 for a description of date-time arguments.
/ALLFILES: YES or NO	Queue operation	Accepts the request only if all of the files in the request exist. By default, if any of the files do not exist, the others will be processed appropriately. This switch specifies that if any file is not present, no files should be processed. The value of YES or NO is optional. If you specify YES, all of the files you specified must exist, or none of the files will be punched.
/ASINCE: date-time	File control	Queues only the files that have been accessed since the specified date and time.
/BEFORE: date-time	File control	Queues only the files with creation dates before the specified date and time. Refer to Section 1.8.3 for a description of date-time arguments.
/CHARACTERISTIC: arg	Queue operation	Specifies an output characteristic. You can find a list of the characteristics arguments defined for your system in the file SYS:CHARTY.DAT.
/CHECK	Queue operation	Prints on your terminal a list of the queue entries made by your job.
/COPIES:n	File control	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 entries.
/CREATE	Queue operation	Makes a new entry in the tape punch queue. This switch is the default function, except when you are listing the entries in the queue.

/DEFERRED	Queue operation	Causes deferred output to be released to the paper tape punch queue. You must use one of the following switches with /DEFERRED: /CREATE completes all released output requests. /KILL eliminates the released output requests. Refer to the SET DEFER command for more information and examples.
/DELETE	File control	Deletes the file after spooling. (This is the same as /DISPOSE:DELETE.)
/DESTINATION: node	Queue operation	Specifies the node that will process the request. Use this switch to specify that the paper tape punch be connected to the specified node. Use the node name or node number to specify the node. Files cannot be punched at a host other than the host node to which your terminal is connected. If the request is made to punch files at another host, or at a node that does not exist, the request will wait in the queue indefinitely.
/DISPOSE:arg	File control	Controls the disposition of the file after it is queued. The arguments to this switch are: DELETE deletes the file from your directory after punching it. PRESERVE keeps the file in your area after punching it. This is the default function. RENAME renames the file into the spooling area. This deletes the file from your directory area immediately.
/DISTRIBUTION: "text"	Queue operation	Specifies text to place in the distribution field, on the banner page of output. You can use this field to include mailing information, or the location where the operator should leave the listing. The text field may be up to 39 alphanumeric characters, including punctuation and spaces if the text is placed in quotation marks.
/ERBINARY	File control	Prints an error message if a binary file is included in the request. This is the default function.

/ERNONE	Queue operation	Prints an error message if no files match the wildcard construction. This is the default function.
/ERPROTECTION	Queue operation	Prints an error message if processing the request requires a violation of file protection. This is the default function.
/FAST	Queue operation	Prints a list of the entries in the paper tape punch queue in a fast format. This is the same as /LIST:FAST.
/FILE:arg	File control	Specifies how the file format is to be interpreted. The following arguments can be used with this switch: ASCII interprets the file as ASCII text. ELEVEN interprets the file as four 8-bit bytes in each 36-bit word. The bits are arranged as follows: Byte 1: bits 10-17 Byte 2: bits 2-9 Byte 3: bits 28-35 Byte 4: bits 20-27
/FORMS:arg	Queue operation	Processes the file on the specified kind of paper tape. The default is NORMAL. Available forms are listed in SYS:FORMST.DAT.
/GENERIC	Queue operation	Sends output to the next available paper tape punch. This is the default function. This switch is the complement to /UNIT.
/HEADER: YES or NO	File control	Makes header units before each file if you type /HEADER or if you specify YES. Does not make headers between files if you specify NO. 1 may be used for YES; 0 may be used for NO. /HEADER:NO is the same as /NOHEADER. /HEADER is the default function.
/HELP:arg	Queue operation	Prints information on your terminal about the QUEUE command. This switch does not queue any files. This switch can be used alone or with one of the following arguments: TEXT prints a message with the format and switches to the QUEUE command. This is the same as /HELP with no arguments. SWITCHES prints a list of all the switches available with the QUEUE command.

/JOBNAME:name	Queue operation	Specifies the name of the job. The job name can be up to six alphanumeric characters.
/KILL	Queue operation	Removes the specified entry from the queue. You must give a job name, /SEQUENCE, or /REQUESTID, to the left of the equal sign in the command line.
/LENGTH:n:m	File control	Processes only files whose length is between n and m blocks.
/LIMIT:n	Queue operation	Limits the output to the specified number of feet.
/LIST:arg	Queue operation	Prints information about the jobs in the queue. If you use /LIST alone, it shows the jobs in the queue. This is equivalent to using the QUEUE command with no arguments and no switches. /LIST can be abbreviated to /L . The switch can take one of the following arguments: ALL shows all data about each queue request. FAST shows a fast list of the queue requests. (This is the same as /FAST.) JOBS shows a list of the jobs in the queue. (This is the same as /LIST with no arguments.) SUMMARY shows only the summary line of the queue display.
/MESSAGE: arg	Queue operation	Specifies the amount of information to be printed when an error occurs from the request. You can specify one or more of the following arguments: ADDRESS prints the location in memory where the error occurred. CONTINUATION prints information about the error. FIRST prints the one-line error message. PREFIX prints a six-character error prefix to the error message.

/MODIFY	Queue operation	Alters the specified parameter in the specified job. This switch requires that you have access rights to the job. You must give a job name, /SEQUENCE, or /REQUESTID, to the left of the equal sign in the command line. This switch can be used to modify a previously submitted request as long as the request has not been started.
/NEW:YES or NO	File control	Accepts the request even if the file does not yet exist.
/NOHEADER	File control	Suppresses header units before file. /HEADER is the default function.
/NONEW	File control	Forces the system to use an existing file. This is the default function.
/NONOTIFY	Queue operation	Does not set the system to notify you when the request is finished. Refer to /NOTIFY.
/NONULL	Queue operation	Prints an error message if none of the files in the request exist. This is the default function.
/NOOPTION	Queue operation	Suppresses the SWITCH.INI file. SWITCH.INI files are described in Appendix B.
/NOPHYSICAL	File control	Recognizes logical names for devices in the command string. This is the default function.
/NOSTRS	File control	Does not scan each structure for files of the same name. This is the default function.
/NOTES:"text"	Queue operation	Punches the text in the header units. The text can be up to 12 characters, and it must be enclosed in quotes if it contains any nonalphanumeric characters, such as spaces.
/NOTIFY: YES or NO	Queue operation	Notifies you on your terminal when request is completed. To be notified, use /NOTIFY with no argument, or with YES or 1 as an argument. To suppress notification, use /NOTIFY:0, /NOTIFY:NO, or /NONOTIFY. By default, you are not notified when a request is finished. In special cases, such as the output of deferred requests, you will never be notified.

/NULL:YES or NO	Queue operation	Does not print a fatal error message if the specified file does not exist.
/OKBINARY	File control	Accepts files whose extensions indicate that they include binary information. Normally, files with extensions .SAV, .SHR, .LOW, .REL, .EXE, and .HGH will not be accepted for processing.
/OKNONE	Queue operation	Does not print a warning message if no files match the wildcard construction.
/OKPROTECTION	Queue operation	Does not print error messages when a protection error occurs.
/OPTION:option	Queue operation	Uses the option line QUEUE:option in the SWITCH.INI file. SWITCH.INI files are discussed in Appendix B.
/PHYSICAL	File control	Does not recognize logical names for devices in the command line.
/PRESERVE	File control	Saves the control file after processing it. This is the default and it is the same as /DISPOSE:PRESERVE.
/PRIORITY:n	Queue operation	Gives the specified priority number (n is 1 to 63) to the request. A larger number has greater priority.
/PROTECTION: nnn	Queue operation	Specifies a protection code for the queue request. Queue requests are protected in the same way that files are protected. Refer to Section 1.9.4.
/REMOTE	Queue operation	Prints on your terminal a list of remote queues. Must be used with /DESTINATION.
/REQUESTID:n	Queue operation	Specifies the request identification number of the job you wish to modify or terminate.
/RUN:file	Queue operation	Executes the specified file after the job is done.
/RUNCORE:n	Queue operation	Executes the file specified in /RUN in nK of core after the job is done.
/RUNOFFSET:n	Queue operation	Executes the file specified in /RUN with offset n after the job is done.
/SEQUENCE:n	Queue operation	Specifies a sequence number to help identify a request to be modified or deleted.

/SINCE: date-time	File control	Queues only the files with a creation date after the specified date and time.
/STRS:YES or NO	File control	Searches for the file on all structures in the search list and takes every occurrence. The default is to take just the first occurrence of the file.
/TAPE:arg	File control	Punches paper tape in the specified mode. If you do not use this switch, the tape is punched according to the data mode specified in the file. You can use any one of the following arguments with this switch: ASCII punches the tape in ASCII mode. BINARY punches the tape in binary mode. IBINARY punches the tape in image binary mode. IMAGE punches the tape in image mode.
/TMPFIL: file:text	Queue operation	Creates a temporary file TMP:file and enters the text into the file.
/UNIT:n	General	Specifies the unit number of the device to which you want the output sent.
/USERNAME: "name"	Queue operation	Specifies the user name field of the banner page of output. This field can contain up to 39 alphanumeric characters, and may include punctuation and spaces if the name is placed in quotation marks.

Associated Messages

When a new entry is made in a system queue, the system prints a message on the user's terminal. The message is in the form:

[PAPERTAPE JOB name QUEUED, REQUEST #nnn, LIMIT xxx]

Where: **name** is the name of the job in the queue. This can be specified by the user. Otherwise, it defaults to the name of the first file in the request.

nnn is the number that represents the request identification of the job in the queue.

xxx is the maximum number of feet that the job will use.

Characteristics

Leaves your terminal in monitor level.

Destroys your core image.

Does not require LOGIN if a list of queue entries is desired.

Example

Punch 3 copies, in binary mode, of the file DSK:SENDMP.REL.

```
.TPUNCH SENDMP.REL/TAPE:BINARY/COPIES:3<RET>
[PAPERTAPE JOB SENDMP QUEUED, REQUEST #131, LIMIT 10]
```

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 1.9.

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

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 (MTxn) or a DECtape (DTxn). Device names are discussed in Section 1.9.1.

Characteristics

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. If your job is currently doing I/O to a file, then CTRL/T will disclose that status, unless the job is execute-only. 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
- current context
- program counter (PC)

Refer to Section 1.6.9 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.

Formats

CTRL/T

USESTAT

Characteristics

CTRL/T:

Leaves your terminal at the same level from which you issued CTRL/T.

Does not echo on your terminal.

Does not interrupt program execution.

Does not disclose file information for execute-only jobs.

The USESTAT Command:

Leaves your terminal at monitor level.

Does not interrupt program execution.

Does not disclose file information for execute-only jobs.

Examples

1. Use CTRL/T for a job that is not execute-only. Note that although it is shown here, CTRL/T does not echo on your terminal.

```
.R PIP<RET>
*COP FILES.ALL=*.*<RET>
<CTRL/T>
Day:13.53 Run:  0.03 Rd:2 Wr:1 PIP 4+10P Ctx:1 RN PC:403171
Input wait for DSKB0:SWITCH.BAK[10,5763] block 1
```

2. Use USESTAT for an execute-only job:

```
.USESTAT<RET>
Day:  29.60 Run:  0.61 Rd:3506 Wr:1 DIR 15+44P Ctx:1 ^C
PC:423003
```

.

VERSION Command

Function

The VERSION command prints on your terminal 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.
[+ high + high]	A SEGOP. monitor call has been done to bring a high segment into core without destroying the existing high 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
```

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.

o

Format

VERSION

Characteristics

Leaves your terminal at monitor level.

Does not destroy your job's core image.

Examples

1. Run the TECO program.

```
.R TECO<RET>
```

Halt the program.

```
*^C
```

Find version number.

```
.VERSION<RET>
TECO 23B(162) +
```

Continue the program and exit.

```
.CONTINUE<RET>
*EX<ESC><ESC>
```

.

2. Type a file.

```
.TYPE WONDER.ALG<RET>
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<RET>
PIP 33B(260) +
```

Run and halt the SYSTAT program.

```
.R SYSTAT<RET>

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<RET>
SYSTAT 472(156)
```

WHERE Command

Function

The WHERE command allows you to determine the system or network node to which a specified device is physically connected. The information given includes:

- o node name
- o node number (ANF-10 and DECnet)
- o software-identification (ANF-10 only)
- o creation-date of software (ANF-10 only)
- o server location string (LAT terminal only)

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 form depends on the type of node. Formats are:

Node type	Format
Local	name number system ID/creation date device line no.
ANF-10	node name number system ID/creation date device line no.
DECnet	name number device (NRT) or (CTERM)
LAT	name location-string device

If the system does not recognize the device you specified, it prints the message:

?NO SUCH DEVICE

Characteristics

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>
Local KL1026(26) R2364A KL # 1026 mm-dd-yy LPT262

.
```

This example shows that the node name is 1026. System identification is RZ364A KL # 1026/1042. Creation date of software is mm-dd-yy. The specified device is LPT262.

2. To find the physical node of your job's terminal, type the command:

```
.WHERE TTY:<RET>
Local NOVA(31) DN87S V##(##) mm-dd-yy TTY70 LINE #70

.
```

This example shows that the terminal is connected to node NOVA(31). The version number is V##(##). The creation date was mm-dd-yy. 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>
Local KL1026(26) RZ345A KL # 1026/1042 mm-dd-yy TTY4 LINE #4

.
```

4. To display the location of the system controlling terminal, type:

```
.WHERE CTY:<RET>
Local KL1026(26) RZ345A KL # 1026/1042 mm-dd-yy 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

Leaves your terminal at monitor level.

Destroys your core image.

Examples

1. Delete the files on DECtape 4.

```
.ZERO DTA4:<RET>
.
```

2. Delete all files in your search list.

```
.ZERO DSK:<RET>
.
```

3. Delete all files in your SFD named CBDDT.

```
.ZERO DSKB:[27,4072,CBDDT]<RET>
.
```

APPENDIX A

FUNCTIONAL GROUPS OF COMMANDS

Table A-1 lists the operating system commands, divided into groups by function. 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 2.1.1 through 2.1.10.

NOTE

Table A-1 lists the commands that are described in Section 2.2. Do not use the commands and programs listed in Table A-1 until you are familiar with them.

Table A-1: Functional Groups of Commands

Command	Functional Description
Job-Control Group	
ATTACH	Connects your terminal to a previously DETACHed job.
CONTEXT	Allows you to create, delete, and switch between contexts.
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.
PASSWORD	Allows you to change your password.
POP	Destroys an inferior context.
PUSH	Creates an inferior context.
REATTACH	Transfers your job from one terminal to another.
SESSION	Changes the account string of your job.

Table A-1: Functional Groups of Commands (Cont.)

Command	Functional Description
Information Group	
ACCOUNT	Prints the account string for your job.
CONTEXT	Displays the status of a context.
CORE	Prints the amount of core memory assigned to your job.
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.
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.
PUNCH	Prints a list of the jobs in the default punch queue.
QUEUE	Prints a list of all the jobs in all the queues.
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.

Table A-1: Functional Groups of Commands (Cont.)

Command	Functional Description
Information Group (Cont.)	
SHOW ALLOCATION	Prints the resources that are allocated to your job, and the structures and tape volumes that are mounted for your job.
SHOW QUEUES	Prints queue requests in each queue.
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.
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 Group	
SET TERMINAL TERMINAL SET TTY TTY INITIA	Sets the specified parameters for your terminal.
Terminal Communication Group	
MAIL	Sends, receives, and organizes messages to and from other users.
PLEASE	Allows communication with the computer operator.
SEND	Sends a message to a specified terminal.

Table A-1: Functional Groups of Commands (Cont.)

Command	Functional Description
File-Handling Group	
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.
DELETE	Removes the specified file from your directory.
DIRECTORY	Prints a list of the files in your directory.
EOF	Writes an end-of-file mark on magnetic tape.
FILE	Transfers files from disk to DECtape and DECtape to disk.
LIST	Prints 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.
PUNCH	Enters files in the default punch queue.
QUEUE	Enters files into the specified queue.
RENAME	Changes the name of a file.
SET DEFAULT PROTECTION	Changes your default protection code.
SUBMIT	Enters files in the batch input queue.
TECO	Calls the TECO text editor to edit an existing 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.

Table A-1: Functional Groups of Commands (Cont.)

Command	Functional Description
Device-Handling Group	
ALLOCATE	Allocates resources to your job for future mounting.
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.
CANCEL	Kills queue requests.
CLOSE	Closes the open files on a device.
CPUNCH	Enters files in the card-punch queue.
DEALLOCATE	Removes resources from your job's allocation list.
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.
MOUNT	Requests ownership of a device.
PRINT	Enters files in the line-printer queue.
PUNCH	Enters files in the default punch 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 DEFER	Defers queueing of output requests until you log out.
SET DENSITY	Sets the default density of the magnetic tape.
SET RETRY	Controls error-reporting from magnetic tapes.

Table A-1: Functional Groups of Commands (Cont.)

Command	Functional Description
Device-Handling Group (Cont.)	
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.
Program-Preparation Group	
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.
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.
DEPOSIT	Puts data into your memory area.
EOF	Writes an end-of-file mark on magnetic tape.
EXAMINE	Displays the contents of the specified core address.
FUDGE	Creates a library .REL file from a temporary file.
GLOB	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.
SET BREAK	Establishes an interrupt point at a specified location in your program.
TECO	Calls the TECO text editor to edit an old file.

Table A-1: Functional Groups of Commands (Cont.)

Command	Functional Description
Program-Control Group	
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.
LOAD	Runs the loader, and loads the .REL file into memory.
MERGE	Loads a second program into memory, to be run with the first.
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 DDT BREAKPOINT	Enables the DDT breakpoint facility.
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.

Table A-1: Functional Groups of Commands (Cont.)

Command	Functional Description
Program-Control Group (Cont.)	
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.
START	Begins execution of a loaded program.
Network Group	
Command	**Functional Description**
ASSIGN	Allows you to access a device on another system in the network.
LOCATE	Changes the default device list of your job.
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 SWITCH.INI file in your UFD into which you can put switches for certain programs. Some of the programs that read SWITCH.INI are:

BACKUP	CREDIR
DIRECTORY	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 UFD. 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 DIRECTORY or LOGIN or a command name which runs one of the programs, such as PRINT. You must use the full program name or at least six characters of any program name that is longer.

/switch is a valid switch for the named program.

Example

```
DIRECTORY/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:

```
DIRECTORY/FAST
```

Then, when you issue the following command:

```
.DIR<RET>
```

The system will print a fast (shortened) listing of your directory. You could override the SWITCH.INI file by typing other switches after the command. For example:

```
.DIR/NORMAL<RET>
```

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 DIRECTORY, 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:

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

If you then type the DIRECTORY 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 DIRECTORY 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:

```
DIRECTORY/OPTION:THISRUN<RET>
```

Examples

1. This example shows one way to use a SWITCH.INI file to set your terminal type automatically when you log in.

   First, log in to the system:

   ```
   .LOG 27,10024<RET>
   Job 36  RLT19L KL1026 AP Monitor  TTY410
   Password: <RET>
   [LGNLAS Last access to [27,10024] succeeded on  6-May-88:14:00:32]
   15:09    6-May-88   Friday
   ```

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.

```
.I TTY<RET>

RLT19L KL1026 AP Monitor 15:05:42 TTY410 system 1026/1042/1322
Connected to Node KL1026(26) Line # 410
Job 35   User DOTY       [27,10024]

TYPE:TTY          APC:HARDWIRED   ECHO            WIDTH:80
LENGTH:0          NOSTOP          FILL:0          LC
NOTABS            NOFORM          CRLF            GAG
NOSBELL           NODISPLA        NOTAPE          BLANKS
NOALTMOD          NORTCOMP        NOREMOTE        XONXOF
NOUNPAUS          NOESCAPE        NOEIGHTB        NOQUOTE
IDLEDI:0          NOSLAVE
```

Create a file named SWITCH.INI, with a command to LOGIN that sets your terminal type.

```
.SOS SWITCH.INI<RET>
Input: SWITCH.INI

00100 LOGIN/TERMINAL:TYPE:VT240<RET>
00200 <ESC>
*ES<RET>
```

Log out:

```
.KJOB<RET>
Job 36  User DOTY  [27,10024]
Logged-off TTY410  at 15:09:53  on  6-May-88
Runtime: 0:00:00, KCS:12, Connect time: 0:00:51
Disk Reads:11, Writes:23
```

Then log in once more.

```
.LOG 27,10024<RET>
Job 36  RLT19L KL1026 AP Monitor  TTY410
Password:<RET>
[LGNLAS Last access to [27,10024] succeeded on  6-May-88:15:00:42]
15:09    6-May-88   Friday
```

Type the command INITIA TTY to see your terminal characteristics.

```
.I TTY<RET>

RLT19L KL1026 AP Monitor 15:10:07 TTY410 system 1026/1042/1322
Connected to Node KL1026(26) Line # 410
Job 36   User DOTY       [27,10024]

TYPE:VT240        APC:HARDWIRED   ECHO            WIDTH:80
LENGTH:24         NOSTOP          FILL:0          LC
TABS              NOFORM          CRLF            GAG
SBELL             DISPLA          NOTAPE          BLANKS
NOALTMOD          NORTCOMP        NOREMOTE        XONXOF
NOUNPAUS          NOESCAPE        NOEIGHTB        NOQUOTE
IDLEDI:0          NOSLAVE
```

2. This example shows a SWITCH.INI file which contains a variety of commands and switches.

```
.TYPE SWITCH.INI<RET>

LOGIN     /TERMINAL:(TYPE:VT240,NOGAG,DEFER,XONXOFF,NOSBELL,-
          PAGE:0)/NOTICE:SOMETIMES/MAILCHECK-
          /RUNOFFSET:1

QUEUE     /NOTIFY/HEADER:NO-
          /LIST:ALL/LOWERCASE/OUTPUT:ERROR/PRESERVE

DIRECT    /NOTEMP

INITIA    /SETTTY/TERMINAL:TYPE:VT100
```

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 deciphers 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<RET>
```

You can then later type the following:

```
.TECO<RET>
```

instead of the command line:

```
.TECO PROGX.MAC<RET>
```

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.

C.1 INDIRECT COMMANDS (@ CONSTRUCTION)

Most programs receive input in the form of commands from the terminal. However, it is possible to write a program that accepts commands from a file. That file is known as an indirect command file.

For example, 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. 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.

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
```

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 override 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, BLISS, or MACY11 as the compiler. However, these switches and compilers are not supported.

The switches that change the default compiler can be **temporary** or **permanent**. (Refer to Section 1.8.4.)

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:

- Group the switches according to each related source file, if you use the + construction.
- 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
- Separate the groups of switches by commas for each source file.
- Enclose all switches for each source file within one set of parentheses.
- 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:

- The switches immediately before a closing right parenthesis are source (SSSS).
- The switches before the first comma are binary (BBBB,,).
- The switches before the second comma are listing (,LLLL,).
- The order of the switches is BBBB, LLLL, SSSS.
- The individual switches are separated by spaces.
- The switches contain no more than 150 characters in all.
- The switches contain only the following non-alphanumeric characters within the parentheses: colon (:), hyphen (-), slash (/), comma (,), and parentheses.

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
```

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.
			S	Suppress the listing of the source program.
COBOL	A	A	A	Allow the listing of code generated.
		C		Produce a cross-referenced listing of all user-defined items in the source program.
	E	E	E	Check the program for errors, but do not generate code.
	I			Suppress generation of the starting address of a main program.
	J			Force a starting address to be generated for a subprogram.
			L	Use the preceding file descriptor as a library file whenever the COPY verb is encountered.

Table C-1: Processor Switches (Cont.)

Processor	Binary	Listing	Source	Meaning
	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.
	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.
		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			Perform optimization of global symbols when producing processor code.
			S	Perform compilation, checking for syntax errors only.

Table C-1: Processor Switches (Cont.)

Processor	Binary	Listing	Source	Meaning
MACRO	A	A	A	Advance the magnetic tape reel by one file.
	B	B	B	Backspace the magnetic tape reel by one file.
		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.

The following are examples of the use of switches:

Examples

.DEBUG TEST(,N,)	Suppress error message during assembly.
.COMPILE OUTPUT=MTA0:(M,W)/L	Rewind the magtape (W), compile the first file, and include the MACRO coding in the output listing file (M). Output files are given the names OUTPUT.REL and OUTPUT.LST.
.COMPILE/MACRO A=MTA0:(,Q,W)/L	Rewind the magtape (W), compile the first file, and suppress Q (questionable) error indications in the listing. When a binary switch is not present, the delimiting comma must appear.
.COMPILE/MACRO A=MTA0:(,Q,)/L	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.
.COMPILE FOO (NOWARN DEBUG)	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 TOPS-10 LINK 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 TOPS-10 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.
B10	Source	Source file in BLISS-10.
B11	Source	Source file in BLISS-11.
B16	ASCII	Source file in BLISS-16.
B32	ASCII	Source file in BLISS-32.
B36	ASCII	Source file in BLISS-36.
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).
BCP	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.

Table D-1: File Name Extensions (Cont.)

File Name Extension	Type of File	Meaning
CKP	Binary	Checkpoint core image file created by COBOL operating system.
CHN	Object	CHAIN file.
CMD	ASCII	Command file for indirect commands (@ construction for COMPIL).
CMP	ASCII	Complaint file by GRIPE.
COR	ASCII	Correction file for SOUP.
CRF	ASCII	CREF (cross-reference) input file.
CTL	ASCII	Batch control file.
DAE	Binary	Default output for DAEMON-taken core dumps.
DAT	ASCII, Binary	Data (FORTRAN) file.
DBS	Binary	Data Base Management System data 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.
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.
LSQ	ASCII	Queue listing created by QUEUE program.
LST	ASCII	Listing data created by assemblers and compilers.

Table D-1: File Name Extensions (Cont.)

File Name Extension	Type of File	Meaning
MAC	Source	Source file in MACRO language.
MAN	ASCII	Manual (documentation) file.
MAP	ASCII	Loader or LINK-10 map file.
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.
P11	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).
PAS	Source	Source file in Pascal language.
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 producting 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.

Table D-1: File Name Extensions (Cont.)

File Name Extension	Type of File	Meaning
RNS	ASCII	RUNOFF input for text file of standards.
RSP	ASCII	Script response time log file.
RSX	All	Files for RSX-11D.
RTB	Object	Read-in mode (RIM10B) format file (PIP).
SAI	Source	Source file in SAIL language.
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).
SIM	Source	Source file in SIMULA language.
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.

Table D-1: File Name Extensions (Cont.)

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).

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:).
1,7	System Accounting PPN.
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:).
5,24	DBS library (device DBS:).
5,25	MIC library (device MIC:).
5,27	CTL library (device CTL:).
5,30	Games library (device GAM:).
5,32	DAS60 software (device D60:).
5,33	Users Test Package (device UTP:).
5,36	LN01 font library (device FNT:).
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:).

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
ACT:	Accounting library	[1,7]	System
ALG:	ALGOL library	[5,4]	System
ALL:		User's	All currently mounted structures
APL:	APL library	[5,3]	System
BAS:	BASIC library	[5,1]	System
BLI:	BLISS library	[5,5]	System
COB:	COBOL library	[5,2]	System
CTL:	CTL library	[5,27]	System
D60:	DAS60 software	[5,32]	System
DBS:	DBS library	[5,24]	System
DMP:	DUMP library	[5,21]	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
FFA:	Operator's area	[1,2]	System
FOR:	FORTRAN library	[5,6]	System
FNT:	Fonts for LNO1	[5,36]	System
GAM:	Games library	[5,30]	System
HLP:	HELP library	[2,5]	System
LIB:	User defined library	Set by each user	System
MAC:	MACRO library	[5,7]	System
MIC:	MIC library	[5,25]	System
MFD:	UFD library	[1,1]	System
MUS:	Music library	[5,16]	System
MXI:	PDP-11 library	[5,3]	System
NEL:	NELIAC library	[5,20]	System
NEW:	New system library	[1,5]	System
OLD:	Old system library	[1,3]	System
POP:	POP2 library	[5,22]	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
SSL:	System Search List	[*,*]	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
UMD:	Field Service library	[6,6]	System
UNV:	MACRO universal library	[5,17]	System
UTP:	User's Test Package	[5,33]	System
XPN:	Crash file 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.

Table E-1: ASCII Card Codes (Cont.)

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	~	176	11-0-1
?	77	0-8-7	DEL	177	12-9-7

NOTE

The ASCII characters } and ~ (octal 175 and 176) are treated by the monitor as ALTmode which is often considered to be the same as ESCAPE.

Table 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.

Table F-1: Temporary Files (Cont.)

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 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] <ESC> 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.
nnnFA1.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.
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.
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.

Table F-1: Temporary Files (Cont.)

Name	Meaning
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.
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

-D-

-E-

READER'S COMMENTS

Your comments and suggestions help us to improve the quality of our publications.

For which tasks did you use this manual? (Circle your responses.)

(a) Installation (c) Maintenance (e) Training
(b) Operation/use (d) Programming (f) Other (Please specify.) _____

Did the manual meet your needs? Yes ☐ No ☐ Why? _____

Please rate the manual in the following categories. (Circle your responses.)

	Excellent	Good	Fair	Poor	Unacceptable
Accuracy (product works as described)	5	4	3	2	1
Clarity (easy to understand)	5	4	3	2	1
Completeness (enough information)	5	4	3	2	1
Organization (structure of subject matter)	5	4	3	2	1
Table of Contents, Index (ability to find topic)	5	4	3	2	1
Illustrations, examples (useful)	5	4	3	2	1
Overall ease of use	5	4	3	2	1
Page Layout (easy to find information)	5	4	3	2	1
Print Quality (easy to read)	5	4	3	2	1

What things did you like *most* about this manual? _____

What things did you like *least* about this manual? _____

Please list and describe any errors you found in the manual.

Page	Description/Location of Error
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions for improving this manual: _____

Name _____ Job Title _____
Street _____ Company _____
City _____ Department _____
State/Country _____ Telephone Number _____
Postal (ZIP) Code _____ Date _____

Fold Here and Tape

Affix
Stamp
Here

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
200 FOREST STREET MR01-2/L12
MARLBOROUGH, MA 01752-9101

Fold Here