decsystem10

OPERATING SYSTEM COMMANDS MANUAL

digital

decsystem10

OPERATING SYSTEM COMMANDS MANUAL

AA-0916C-TB

August 1977

This manual reflects the software of the 6.03 release of the monitor.

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754.

digital equipment corporation · maynard. massachusetts

First Priinting,	May 1971
Revised:	December 1971
Revised:	June 1972
Revised:	April 1973
Revised:	May 1974
Revised:	July 1975
Revised:	August 1977

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.

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

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

CONTENTS

CONTENTS (CONT.)

CONTENTS (CONT.)

CONTENTS (CONT.)

FIGURES

TABLES

PREFACE

The DECsystem-10 Operating System Commands manual is a complete reference document describing the commands available in the DECsystem-10 Operating System. Information presented here reflects the 6.03 release of the monitor and related programs. Commands to the monitor or command language interpreters are presented in alphabetical order for easy reference.

DECsystem-10 Operating System Commands does not include reference material for assembly language programming. This information may be found in DECsystem-10 Monitor Calls (AA-0974C-TB). The Monitor Calls manual is intended for experienced assembly language programmers. It includes discussions of monitor programmed operators and the I/O devices connected to the system.

The Introduction to DECsystem-10 manual (DEC-10-MZDC-D) gives an overview of the DECsystem-10 and its operating system, TOPS-10. This manual is written for someone familiar with computers and computing concepts who wishes to understand the functions of DECsystem-10 components. It is not intended to be a programmer's reference manual and should be read once before reading the preceding documents.

SYNOPSIS OF DECSYSTEM-10 OPERATING SYSTEM COMMANDS

Chapter 1 presents all available commands and introduces the operating system components that interface with the user. Chapter 2 is a detailed description of commands the monitor command language interpreter processes. Chapter 3 describes the system error messages and error codes, with correction procedures. The appendices contain all supplementary reference material and tables.

CONVENTIONS USED IN DECSYSTEM-10 OPERATING SYSTEM COMMANDS

The following conventions are observed within this document:

dev: Any logical or physical device name. A colon (:) must be included when a device is specified as a part of a file specification.

list A single file specification or a string of file specifications, consisting of a file name with or without a file name extension; device name if the file is not on disk; project-programmer number if the file is not in your disk area; protection code.

arg A letter, a word, or a word abbreviation specifying the desired function of the command.

jobn Any job number assigned by the monitor.

file.ext Any legal file name and extension.

core The decimal number of 1K blocks containing 512-word pages of memory.

addr An octal address.

c(addr) Contents of an octal address.

[proj,prog] Project-programmer numbers; the square brackets ([ ]) must be included in the command string.

fs Any legal file structure name or abbreviation.

$ The symbol indicating when you type an ESCape or ALTmode.

^X A control character obtained when you press the CTRL key and type a character key.

* The system program response to a command string.

. The monitor's response to a command string.

↵ The symbol used to indicate when you should press the RETURN key. You should type a carriage return after every command to the monitor command language interpreter. At times you can also terminate commands with an ESCape, ALTmode, CTRL/L, etc.

______ Underscoring indicates user input, i.e., the words and numbers that you are to type.

n A decimal number.

= An equal sign, which is used in command strings to separate input and output file specifications.

CHAPTER 1

INTRODUCTION

The DECsystem-10 Operating System, called TOPS-10, is the interface between you and the actual machine. Some of its functions are listed below.

1. Scheduling multiple and simultaneous uses of the system

2. Protecting system users from one another

3. Allowing access to system resources, including peripheral devices

4. Providing a comprehensive disk file system

5. Directing data flow between peripheral devices and your programs

6. Controlling non-interactive jobs

7. Overlapping input/output operations with computations, for high system efficiency

You communicate with the operating system by means of the monitor command language. With the command language, you may access available resources of the computing system and obtain services provided by the operating system.

1.1 JOBS

The DECsystem-10 is a multiprogramming system; 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 initiate from your terminal or card deck, or the operator initiates from a console. You initiate a job from your terminal by using the LOGIN command, and you end your job by using the KJOB command. If you initiate a job with a card deck, the job begins with the $JOB card and ends with the $EOJ (end-of-job) card. Operator jobs begin when the operator initializes the system, and they end when the system is taken down.

You may initiate a job at the central computer site or from remote, telephone-connected, locations whether those jobs are batch, timesharing, or real-time. Once you initiate a job, it is possible for you to also initiate another job without killing the first. For example, you may initiate a timesharing job and, using the SUBMIT command, also submit a second job for batch processing. (Refer to Chapter 2.) You may then wait for the results from the batch job or have the results automatically output while you continue your timesharing job. When configuring and loading the DECsystem-10, the system administrator sets a maximum number of jobs that the system can handle. This number may be set to 511 jobs if the system has sufficient memory, disk storage, processor capacity, and terminals.

1.2 MONITOR MODE AND USER MODE

Your terminal is either in monitor mode or user mode. In **monitor mode**, each line you type is sent to the monitor command language interpreter. When you execute certain commands, the system places the terminal in user mode. When the terminal is in **user mode**, it simply becomes an input/output (I/O) device for you. Your programs use the terminal for two purposes:

1. To accept command strings from the terminal (user mode)

2. To use the terminal as a direct I/O device (data mode)

The following is an example of terminal dialogue:

```
Monitor mode   .R PIP↵                   Monitor command
User mode      *DSK:PROG.MAC=TTY:↵       User command string
Data mode      THIS IS FILE 1^Z          User program using the
                                         terminal
                                         as an input device

               *^C                       CTRL/C
Monitor mode   .R MACRO↵                 Monitor command
User mode      *,TTY:=DSK:PROG↵          User command string
               .
Data mode      .
               .
               assembly listing
               .
               .
               .
```

The special control character ^C (produced when you type the CTRL key and a character key) allows you to stop your program, returning the terminal to monitor mode. If your program is waiting for input from the terminal, you need to type only one CTRL/C to return the terminal to monitor mode; otherwise, you must type two CTRL/Cs. Because of this, you know that your program is not waiting for input if the system does not respond when you type one CTRL/C. You can type certain commands that will cause your program to run or continue (noted in the following chapter), but these commands will leave the terminal in monitor mode.

When the operator starts the system, each terminal is placed in monitor mode, ready for users to LOGIN to them (unless it is slaved or ASSIGNed by another job). However, if the system becomes fully loaded (i.e., the maximum number of jobs that the system is set to handle has been reached) and you try to LOGIN, you will receive the following error message:

```
JOB CAPACITY EXCEEDED
```

When you receive this message, wait a few minutes before trying to LOGIN again.

If you are using the system in a batch card environment, you may think of your cards as being in input spooler mode, monitor mode, or user mode. When your card is in input spooler mode, the card deck contains a control command beginning with a dollar sign ($) and is sent to the Spooler, SPRINT-10. (Refer to the GALAXY-10 Reference Manual.) SPRINT-10 interprets these commands and performs various actions to create a control file for the Batch Controller. When your card is in monitor mode, it contains a monitor command preceded by a period and is copied by SPRINT-10 into the control file. When the card is in user mode, it contains a user-level program command preceded by an asterisk (*) or an equal sign (=) and is also copied by SPRINT-10 into the control file. As each line in the control file is executed, the Batch Controller passes the monitor-level line to the monitor command language interpreter, and the user-level line to the user program. Figure 1-1 illustrates a sample card deck.

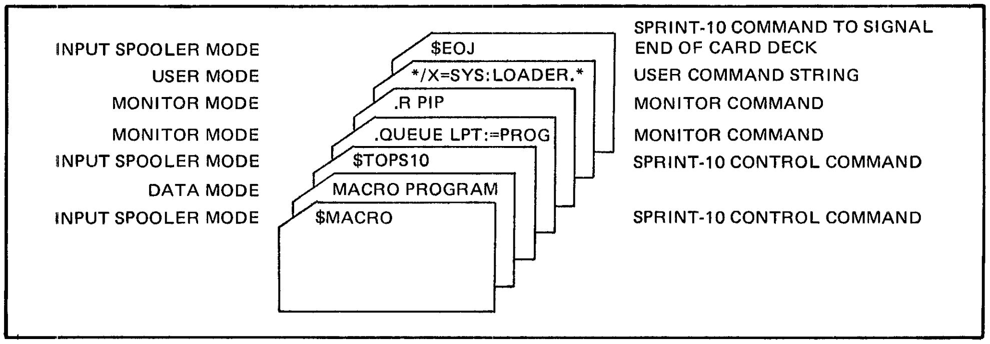


Figure 1-1 Sample Card Deck

1.3 COMMAND INTERPRETERS

1.3.1 Monitor Command Language Interpreter

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

The system, i.e., interpreter, processes most commands without delay. However, the system may momentarily delay a command if your job is swapped out to the disk and the command requires that your job be resident in core memory; the system executes the command when your job is swapped into core. The system signals the completion of each command when it outputs a carriage return/line feed sequence. If the system leaves your terminal in monitor mode, the system prints a period following the carriage return/line feed sequence. If the system leaves your terminal in user mode, the characters that are typed at the terminal come from your program and not from the operating system. Most standard system programs print an asterisk (*) on your terminal, indicating their readiness to accept your command strings.

You may use the type-ahead technique if you are an experienced timesharing terminal user. With the type-ahead technique, you do not have to wait for the completion of one command before you begin another. If you desire two operations from the monitor, you can begin typing the request for the second operation before you receive the period that the system types after completing the first operation.

The command interpreter makes several checks before processing your commands. If you have not logged into the system and you have typed a command that requires that you be logged in, the system responds with the following error message:

```
?LOGIN PLEASE
```

and the system does not execute the command that you typed. The commands described in Chapter 2 require that you be logged into the system unless it is explicitly stated otherwise.

When the system recognizes that you typed a command that requires that your job have core, and your job has no allocated core, the system prints the following error message:

```
?NO CORE ASSIGNED
```

and the system does not execute the command that you typed.

The monitor command language interpreter recognizes several special characters that cause specific functions to be performed. A CTRL/C interrupts your current running program and returns your terminal to monitor mode. This character causes the input line, back to the last break character (e.g., carriage return/line feed), to be deleted. This action is the equivalent of a CTRL/U. You must type two CTRL/Cs if your program is not requesting input from your terminal (i.e., if your program is in the middle of execution).

When you type the DELETE key, the system generates a character that causes the last character you typed to be deleted. This function permits you to correct typing errors. When you press the DELETE key n times, the system deletes the last n characters that you typed. The system echoes all deleted characters on your terminal and encloses the deleted characters in backslashes (\). The system cannot delete any characters that you typed before the last break character or any characters already processed by your program.

When you type a CTRL/U, the system deletes your current input line, back to the last break character. The system responds with a carriage return/line feed, after which you may retype the line. Once you have typed a break character, you can no longer use the line-editing features (e.g., CTRL/U and DELETE) on that line, except when you are using an editor.

When you type a CTRL/O, the system temporarily suppresses output to your terminal. You would find this function useful when a program begins to output a long message that you are not interested in reading. If you do not want to wait for the system to finish printing the message, you can stop the system from outputting the message in one of two ways. The first way is for you to type two CTRL/Cs, but this action also stops the execution of your program. The second way, is for you to type a CTRL/O. When you type a CTRL/O, the system continues executing your program, but it does not print any output on your terminal. The system responds with a carriage return/line feed sequence. The system reinstates output to the terminal when one of the following conditions occurs:

1. Your executing program requests input from your terminal.

2. Your program terminates and returns control to the monitor.

3. You type a CTRL/C, which returns control to the monitor.

4. You type another CTRL/O.

When you type a CTRL/T, the system prints status information pertaining to your job on your terminal. The information that the system prints is the same as it prints when you issue the USESTAT command. This information is listed below.

1. The incremental day time, 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 USESTAT command

2. The incremental run time, which is the CPU time used since you last issued a CTRL/T, USESTAT command, LOGIN command, or TIME command

3. The increment disk reads, which is the disk blocks read since you last issued a CTRL/T, USESTAT command, LOGIN command, or DSK command

4. The program name

5. The core size

6. The job state

7. The program counter, which is the address of the current instruction

8. The I/O state of the file, which can be INPUT WAIT or OUTPUT WAIT

9. The name of the file

10. The block number of the file if the system is performing user I/O, or the words (MONITOR I/O) if the system is performing monitor I/O

11. The number of pages in the current working set, if running under virtual memory

12. The number of pages not in the working set

13. The amount of virtual core currently in use

14. The page rate

By typing a CTRL/T, you can determine your job's progress without interrupting its execution. Note that when you type a CTRL/T the CTRL/T character is not passed to the job as an input character. However, some programs activate a special interrupt feature when you type a CTRL/T. (Refer to DECsystem-10 Monitor Calls.) When you type a CTRL/T, control of such programs automatically transfers to a routine used to output status information, in addition to the regular USESTAT printout.

For example, when you type a CTRL/T while using an editor, which is in the midst of a search, information concerning the progress of the search may be printed. Note that CTRL/T output is typed even if you have suppressed output by use of CTRL/O.

When you type a CTRL/R (^R), the system types the current input line, after it has processed all DELETEs. If you type a line incorrectly, then make corrections using the DELETE key, the system will print the corrected line when you type a CTRL/R. An example of this function is shown below.

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

When you type a CTRL/R, the system issues a carriage return/line feed sequence before printing the corrected input line and leaves the input position indicator at its previous location.

1.3.2 Batch Command Interpreter

All monitor commands that you use via the batch system use the monitor command language interpreter. In addition to monitor commands, batch users can issue commands that are used only by the batch programs: the Input Spooler (SPRINT-10), and the Batch Controller (BATCON). The Input Spooler processes the control commands. (Refer to the GALAXY-10 Reference Manual.)

By means of these commands, you can create a control file, a log file, and data files; you can enter jobs into the batch input queue; you can insert monitor commands into the control file. The Batch Controller performs an additional interpretation. When the system executes your job, the Batch Controller processes the control file to pass monitor commands to the monitor command language interpreter and it passes user-level commands to the appropriate programs.

1.4 COMMAND FORMATS

You type each command as a line of ASCII characters in uppercase or lowercase, or a mixture of uppercase and lowercase characters. If you type spaces or TABs before a command name, the system ignores them. You can type comments on the same line as a command by preceding the comment with a semicolon. The monitor and batch command language interpreters do not interpret or execute a line of comments. You should terminate every command to the monitor command language interpreter by pressing the RETURN key. In this manual, the ↵ symbol indicates when you should press the RETURN key. If you should type an unrecognizable command, the system types the entire line again, preceding and following the line with a question mark. The system leaves the terminal in monitor mode.

You can continue command lines to several system programs (e.g., DIRECT, DUMP, QUEUE, and LINK-10) by placing a hyphen (-) as the last nonblank, noncomment character before the carriage return/line feed sequence (i.e., before pressing the RETURN key). The system treats continuation lines as part of the current command line, which it does not consider as being terminated until it sees a carriage return/line feed sequence that is not preceded by a hyphen. This feature allows you to type indefinitely long command lines.

1.4.1 Command Names

The commands you type to the monitor command language interpreter are one to six characters long. If you type any character past the sixth character, the system ignores it. You need only type enough characters to uniquely identify the command. However, when you are entering a batch job, you should use the command's full name, because the number of unique characters may change when new commands are added to the operating system. If this were the case, your card decks or control files containing abbreviated command names may no longer function properly if new commands were added to the monitor.

Installations that implement additional commands should preserve the uniqueness of the first three letters of all existing commands.

When you issue control commands to the Stacker in the multiprogramming batch system, you must issue a dollar sign ($) in the first column of the card or line and an alphabetic character in the second column. You can abbreviate the command name, as long as the specified command name is unique. The first three characters of the command generally denote uniqueness.

1.4.2 Command Arguments

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

```
?TOO FEW ARGUMENTS
```

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

1.4.2.1 **Project-Programmer Numbers and Passwords** - Only authorized users have access to the DECsystem-10. Your installation's system administrator provides each authorized user with a project number, a programmer number, and a password. The project numbers range from 1 to 377777 (octal). (Note that numbers 1 through 10 are reserved for Digital.) The programmer numbers range from 1 to 777777 (octal). (Note that numbers 1 through 7 are reserved for Digital and numbers 400000 through 777777 are reserved for special purposes.)

NOTE

> When the project number is in the 1 to 10 range, all programmer numbers are reserved for Digital. When the programmer number is in the 1 to 7 range, all project numbers are reserved for Digital.

These numbers identify each user and his corresponding area on file structures. When you specify a project-programmer number in a command string, the project and programmer numbers are separated by a comma and are enclosed in square brackets (e.g., [27,4072]).

Your password is one to six characters long and is used only when you log in to the system. To maintain password secrecy, the monitor does not echo the password when you type it. When you are using a terminal with local copy, the monitor types a mask when you type your password, thereby making the password unreadable. (Refer to DECsystem-10 Monitor Calls.)

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

You can associate a logical device name with a particular physical device. A logical device name can be one to six characters long, followed by a colon. When you refer to the device, you can refer to it by logical name or physical name. Logical device names allow you, when writing programs, to use arbitrarily selected device names and to assign a physical device to the logical name at execution time. However, you should use care when assigning logical device names, because they have priority over physical device names. If you assigned the logical name DSK to a DECtape, every time you attempted to use the disk via the name DSK, you would use the DECtape instead, because of the logical name assignment.

Except for disk devices, you can assign only one logical device name to a physical device at one time. You can use the name for another device by disassociating the name from the first device and then associating it with the second device, via the ASSIGN command. You disassociate logical names from physical devices by using the DEASSIGN command. (Refer to Chapter 2.) Each time you issue a subsequent ASSIGN command , the system replaces the old logical name with the newly specified logical name.

Table 1-1
System Devices

Device	Generic Physical 3-letter	Device Name 2-letter
Card punch	CDP:	CP:
Card reader	CDR:	CR:
Console TTY	CTY:	---
DECtape	DTx:	DT:
Disk	DSK:	DS:
Packs	DPx:	DP:
Fixed head	FHx:	FH:
Display	DIS:	---
Line printer	LPT:	LP:
lowercase/uppercase	---	LL:
uppercase	---	LU:
Magnetic tape	MTx:	---
7-track	---	M7:
9-track	---	M9:

Table 1-1 (Cont.)
System Devices

Device	Generic Physical 3-letter	Device Name 2-letter
Operator terminal	OPR:	---
Paper-tape punch	PTP:	PP:
Paper-tape reader	PTR:	PR:
Plotter	PLT:	---
Pseudo-TTY	PTY:	---
System library	SYS:	SY:
Terminal	TTY:	TT:

The following examples illustrate the use of physical and logical device names:

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

You request a DECtape drive with the logical name ABC.

```
DTA260 ASSIGNED
```

The monitor gives you device DTA260. You then mount a DECtape on drive DTA260.

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

You request the paper-tape punch with the logical name ABC.

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

The monitor types an error message, telling you that the paper-tape punch is reserved.

The logical name ABC now refers to the paper-tape punch.

```
.R PIP
```

You request the system program PIP.

```
*ABC:=DTA260:FILEA
```

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

```
*^C
```

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

```
.ASSIGN DTA:  DEF:
```

You request another DECtape drive with the logical name DEF.

```
ASSIGNED TO JOB N1,N2,...
```

The monitor issues a message telling you that all drives are in use by the specified jobs. The monitor does not assign a DECtape drive or a logical name.

```
.ASSIGN DTA6:  DEF:
```

You request drive DTA6 (which you already have assigned to you) with the logical name DEF. The system clears the copy of the directory currently in core.

`DTA261 ASSIGNED`	You mount a new DECtape on the previously assigned drive. The system reads in the new DECtape directory into core when you next access the DECtape.
`.DEASSIGN PTP:`	You deassign the paper tape punch, thereby clearing the logical name ABC.
`.R PIP`	You request the PIP system program.
`*ABC:=DEF:FILEB`	You request that the file FILEB be transferred from device DEF to device ABC.
`?DEVICE ABC DOES NOT EXIST`	The system prints an error message indicating that the logical device name ABC is no longer assigned.
`*^C`	You type a CTRL/C, returning your terminal to monitor mode.
`.ASSIGN DTA260:  XYZ:`	You request drive DTA260 again with the logical name XYZ. The system disassociates the logical name DEF from DTA260 and clears the directory from core.
`DTA261 ASSIGNED`	You mount a new DECtape. The system reads the new directory into core when you next access the DECtape.

1.4.2.3 **File Structure Names** - In the system, disk devices are grouped according to file structures. File structures are logical arrangements of 128-word blocks on one or more disk units of the same type. A file structure can exist on one disk unit; or, it can be distributed over several disk units of the same type and designated by a single name. However, two file structures cannot exist on the same unit. The operator assigns a SIXBIT name to every file structure when the operator defines the system's file structures. This name is one to five characters long, and cannot duplicate a physical device name, a unit name, or an existing file structure name. The recommended names for public file structures are DSKA, DSKB, ..., DSKn in order of decreasing speed.

1.4.2.4 **File Specifications** - The system stores all information (both programs and data) as named files. Each named file has a file specification associated with it. The file specification consists of the following information:

1. The physical device name or file structure name

2. The file name

3. The file name extension

4. The ordered list of directory names

5. The access protection code

The first four items of the file specification are necessary to identify a disk file uniquely. If you issue a file specification for devices other than DECtape or disk, the system ignores them. DECtapes and disks are the only directory-oriented devices. Note that item 4 in the list above does not apply to DECtapes.

The physical device name used for DECtape or the file structure name used for disk may be any legal device name described in the previous sections of this manual. You should always type a colon following the device name (e.g., DTA3:).

A file name is one to six characters long; the system ignores all characters past the sixth. 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 A for a list of standard file name extensions.) You should use only standard file name extensions, though other extensions may be valid. Note that most programs recognize only file names and extensions consisting of letters and digits.

The ordered list of directory names identifies the disk area where the file is stored. This list may 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. 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, all other users. The standard protection code is 057, allowing users in the owner's project to read and execute the file, and prevent access by all other users. (For a complete description of access protection, refer to DECsystem-10 Monitor Calls.) Note that the standard protection code can be redefined by your installation.

Also, you can issue the SET DEFAULT PROTECTION command. The argument you specify in this command overrides the standard protection code when you create subsequent files.

The following information is needed in command strings when referring to a file:

1. The file name

2. The device name, if the file is not on disk

3. The directory name, if the file is not in your disk area

The following information is optional in a file specification:

1. The file name extension

2. The device name, if the file is on disk

3. The directory name, if the file is in your disk area

4. The protection code

The following are examples of file specifications:

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

1.4.2.5 **Wildcard Constructions** - You can use wildcard constructions with many command strings. A wildcard exists when you type an asterisk to specify a file specification field or a question mark to replace a character in a file specification field. You use the asterisk (*) as a wildcard to designate the entire file specification field. For example:

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

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

filename.M??	All files with this file name and any extension beginning with M
TES??.ext	All files with this extension and any file name up to five characters, beginning with TES
??.???	All files with file names of two characters or less and file name extension of three characters or less.
[25,5??]	All files in directories with the project number 25 and a programmer number 500 through 577

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

??.*	All files with file names of two characters or less

Also, 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 UFD [15,23]

[,30] The UFD that has the your project number and the specified programmer number (i.e., 30)

[36,] The UFD that has the specified project number (i.e., 36) and your programmer number

[,] Your UFD

[-] Your default directory which may be different from Your UFD. (Refer to the SETSRC description in DECsystem-10 Monitor Calls)

[,,SUB1,SUB2] The sub-file directory SUB2 under the sub-file directory SUB1 in your UFD

The DUMP, DIRECT, and QUEUE programs recognize the number sign as described below. (Refer to the PIP Programmer's Reference Manual for PIP's interpretation of the number sign.) You can use a number sign to represent a file name or extension that contains characters that cannot be typed because they have special meanings in the system. For example, if you were to type *.MAC in a command string, you would be referencing all files with the .MAC extension. You can use the number sign followed by the octal representation of the file name or extension in SIXBIT code to allow a file name or extension composed of special characters to be typed. For example, 120000000000 represents the file named *. If letters or digits are part of the file name or the extension containing the special characters, you must also type the octal representation of the letters or digits following the number sign. Therefore, you must type the number sign at the beginning of the file name and all characters following it must be represented in SIXBIT. Appendix F contains a table of SIXBIT character codes.

1.4.2.6 **Date-Time Arguments** - Certain switches require arguments specifying 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; an absolute argument specifies a particular date or time. The format of a relative argument is shown below:

```
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 are required.

minutes are optional.

seconds are optional. However, if you specify seconds, minutes must also be given or seconds will be interpreted as the minutes argument.

You must type the colon to separate one field from the other. You can precede a relative argument with an optional plus (+) or minus (-) sign. If you do specify a 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 three days, four hours, 27 minutes, and 21 seconds ago. Similarly:

```
+4
```

means four hours from now.

An absolute argument is written in the following format:

weekday or date:hours:minutes:seconds

where: weekday is the day of the week and it is optional.

date is optional and has one of the following formats:

day of month-month-year (21-OCT-76)

month-day of month-year (OCT-21-76)

numeric month-day of month-year (10-21-76)

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

hours is based on a 24-hour clock and it is required

minutes is optional

seconds is optional

For example, the following specifies before or since last Wednesday at 9:15:6 A.M.

```
WED:09:15:6
```

Because the date is known to be past or future from either the switch used (e.g., BEFORE and SINCE imply past; AFTER implies the future) or by the plus or minus sign, an unspecified field is filled in so that the result is the next or last occurrence of the specified date. Therefore, 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 P.M.) if future. Therefore:

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

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

/SINCE:JUL-3-76 is since July 3, 1976 at noon

1.4.3 Option Files

You can create files in your disk area into which you can put switches for the DIRECT, DUMP, LINK-10, LOGIN, INITIA, and QUEUE programs. These files allow you to automatically override system defaults for the individual programs.

This file is called DSK:SWITCH.INI and it must reside in your disk area. It can contain two types of lines. The first type is written in the following format.

```
program-name/switch/switch/.../switch
```

where: program-name is either DIRECT or DUMP.

/switch is a valid switch for the named program (i.e., /DIRECT, /DETAIL, /NOSUMMARY).

You can place only one line per program in the file. However, this line may be as long as you wish by putting the hyphen continuation character before the carriage return. When you run the named program, the switches in SWITCH.INI will be used as the defaults instead of any system-defined defaults. When the system calls in the program, it searches for your file called SWITCH.INI. If the system does not find the file, the system uses any system defaults for that program. If the system finds the file, but does not find a line for the specified program, the system uses any system-defined defaults. When the system finds your SWITCH.INI file and it contains a line for the specified program, the system uses the switch values that you have specified in the SWITCH.INI file instead of any system-defined 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 SWITCH.INI.

The second type of line that can appear in your SWITCH.INI file is written in the following format:

```
program-name:option-name/switch/switch.../switch
```

where: program-name is a program name such as DIRECT, DUMP, LINK-10, LOGIN, INITIA, or QUEUE.

option-name is the same as used for the /OPTION switch. (Refer to the DIRECT and DUMP command descriptions in Chapter 2.)

You use the second type of line to override both system defaults and any defaults that you may have previously specified. You reference this type of line in SWITCH.INI by including the /OPTION:option switch in a command string to a program. If you specify an option name in the command string that does not appear in SWITCH.INI, the program outputs a warning message and uses the system defaults for that program.

Assume that you create a file called SWITCH.INI that contains:

```
DIRECT/FAST/UNITS/SUMMARY
DIRECT:THIS RUN/WORDS/ACCESS:25
```

If you then type the DIRECT command, the system will print a fast listing showing both the actual unit names (instead of the structure names) and the summary line. When you desire the system to print a normal directory listing, you must type a command string to DIRECT that includes the /NORMAL switch. Note that disk unit names, not structure names, and the summary line will still be output. 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 words or less:

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

1.5 COMPIL-CLASS COMMANDS

You can type short, concise command strings that cause a series of options to be performed. These commands are known as COMPIL-class commands; they are described in detail in Chapter 2. These commands caüse the monitor to run the COMPIL program, which deciphers the command and constructs new commands strings for the system program (e.g., TECO, PIP, LINED, SOS, FORTRAN) that actually process the command. Each time you execute the CREATE, MAKE, EDIT, or TECO commands, the command with its arguments is written as a temporary file in core or on the disk. Therefore, the system can recall the file specification last edited for the next edit without your having to specify the arguments again. Note that this is an exception to the requirement that the file name must always be specified. For example if you type:

```
.CREATE PROGX.MAC
```

You may then later type the following:

```
.EDIT
```

instead of the following command line:

```
.EDIT PROGX.MAC
```

assuming that you have not issued another EDIT-class command changing the file name in the interim.

The system also writes the COMPILE, LOAD, EXECUTE, and DEBUG commands, with their arguments, in a temporary file, allowing your last file specification to be recalled without again specifying the arguments.

The temporary files containing these file specifications have file names written in the following format:

```
nnnxxx.TMP
```

where: nnn is your job number in decimal, with leading zeros to make three digits.

xxx specifies the use of the file.

(Refer to Appendix C for a list of the temporary files.)

1.5.1 Indirect Commands (@ Construction)

When you have to input many program names and switches, you can put these into a file, eliminating the need for you to retype the names and switches for each compilation. You do this by using the @ file construction, which you can combine with any COMPIL-class command.

You can specify an @ file at any point in a command line after the first word in the command. In this construction, the word **file** must be a file name, followed by an optional file name extension and project-programmer number. If you omit the extension, the system searches for a command file with a .CMD file name extension. If that file is not found, it then searches for a command file with a null extension. Then, when the system finds the specified file, it places the information in the file into the command string, replacing the @filename.

For example, if you have a file called @FLIST containing the following command string:

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

then, the following command line:

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

could be replaced by the following command line:

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

You may have command files containing the @ file construction to a depth of 17 levels. If this process of indirection results in files pointing in a loop, the maximum depth is rapidly exceeded, and the system prints an error message.

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

1. Spaces are used to delimit words, but otherwise are ignored. Similarly, a TAB, a vertical TAB, and form feed characters are treated as spaces.

2. To allow long command strings, command terminators (i.e., carriage return, line feed, ESCAPE) are ignored if the first nonblank character after a sequence of command terminators is a comma. Otherwise, they are treated either as commas by the COMPILE, LOAD, EXECUTE, and DEBUG commands, or as command terminators by all other COMPIL-class commands.

3. Blank lines are ignored completely.

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.

1.5.2 The + Construction

When you use the + construction, the system produces a single relocatable binary file from a collection of input source files. If you wish 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 system produces one binary file produced 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 system gives the name of the last input file in the string to any output file (i.e., .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]
```

1.5.3 The = Construction

Usually the system 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 name of the output files. For example, if you need a binary file called BINARY.REL from a source program called SOURCE.MAC, you could issue the following command line:

```
    .COMPILE BINARY=SOURCE
```

You can also use this construction to specify an output name for a file produced by the use of the + construction. To give the name WHOLE.REL to the binary file the system produces with PART1.MAC and PART2.MAC, you could issue 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 may also use it to change any other default condition. The default condition for processor output is DSK:source.REL[your ppn]. If you would like output on DTA3 with the file name FILEX instead of the defaults, you could issue the following command line:

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

1.5.4 The < > Construction

The < > 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 merely type the following line:

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

The following command string construction is invalid:

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

1.5.5 Compile Switches

You can modify the COMPILE, LOAD, EXECUTE, and DEBUG commands by including switches in the command string. The switches can be used to perform the following:

1. Indicate the processor to be used
2. Force a compilation
3. Generate listings
4. Create libraries
5. Search user libraries
6. Obtain loader maps

You precede each switch with a slash and terminate it with a non-alphanumeric character, usually a space or a comma. You can abbreviate the switch if the name remains unique.

The switches used are either temporary or permanent. A **temporary** switch applies only to the immediately preceding file. You cannot place an intervening 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.

A **permanent** switch applies to all files following it, until you modify it by a subsequent switch. You separate the switch from the file name by spaces, commas, or a combination of both. In the command construction:

```
.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     ;temporary switch that
                                    ;affects PROG
```

The COMPILE, LOAD, EXECUTE, and DEBUG command descriptions in Chapter 2 list their applicable switches.

1.5.6 Standard Processor

The system always translates files with recognizable extensions implied by the extension (i.e., .MAC, .CBL, .F4, .ALG), even if you type a switch directing otherwise. The COBOL compiler will process a file called DATPRO.CBL. Files without a recognizable processor extension are compiled or assembled according to the standard processor, normally FORTRAN, at the beginning of the command string.

You can control the setting of the standard processor 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 descriptions of the switches used to change the standard processor.

In the following examples, the installation has chosen FORTRAN as the standard processor. 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 to be compiled by FORTRAN because MIN is not a recognizable processor 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, SNOBOL, BLISS, and MACY11 will be recognized as processors. However, these assembly switches are not supported.

The switches used to change the standard processor can be **temporary** or **permanent**. (Refer to Section 1.5.5.) The command construction:

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

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

If you specify source files with appropriate extensions, you can disregard the subject of the standard processor. Any files you specify with a processor extension are always translated by the specified processor, even if you type the switch name.

1.5.7 Processor 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
2. Listing file
3. Source file

To include switches with these files, you must do the following:

1. Group the switches according to each related source file if you use the + construction.
2. Group the switches according to the three types of files for each source file; the order of the groups of switches is
   a. binary
   b. listing
   c. source
3. Separate the groups of switches by commas for each source file.
4. Enclose all switches for each source file within one set of parentheses.
5. Place each parenthetical string immediately after the source file to which it refers.

The COMPIL program interprets the groups of switches, using these rules:

1. The switches immediately before a closing right parenthesis are source (SSSS).
2. The switches before the first comma are binary (BBBB,,).
3. The switches before the second comma are listing (LLLL,).
4. The order of the switches is BBBB, LLLL, SSSS.
5. The individual switches are separated by spaces.

The following are 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 1-2, with their meanings and the types of files to which they apply.

The following are examples of the use of processor switches:

.DEBUG TEST(,N,) Suppresses error message during assembly.

.COMPILE OUTPUT=MTA0:(S,M,W)/L Rewinds the magtape (W), compiles the first file, produces binary output for the KI10 (S), and includes 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
```

Rewinds the magtape (W), compiles the first file, and suppresses 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
```

Compiles the file at the current position of the tape and suppresses Q error indications on the listing. Note that when the source and binary switches are not present, the delimitng commas must appear.

```
.COMPILE FOO (NOWARN DEBUG)
```

Compiles the file named FOO using the F10 source switches NOWARN and DEBUG.

Table 1-2
Processor Switches

Processor	Binary	Listing	Source	Meaning
ALGOL	D			Set dynamic storage region for your own array (knwon as the heap). E The source file has line number in columns 73 to 80.
			L	List the source program.
		N		Suppress error message 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.

Table 1-2 (Cont.)
Processor Switches

Processor	Binary	Listing	Source	Meaning
	I			Suppress generation of the starting address of a main program.
	J			Force a starting address to be generated for a subprogram.
			L	Use the preceding file descriptor as a library file whenever the COPY verb is encountered.
	M	M	M	Print a map showing the parameters of the user-defined item.
		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		Zero the DECtape directory.
FORTRAN	A	A	A	Advance the magnetic tape reel by one file.
	B	B	B	Backspace the magnetic tape reel by one file.
		C		Generate a CREF-type cross-reference listing.
		D		List error message codes only.

Table 1-2 (Cont.)
Processor Switches

Processor	Binary	Listing	Source	Meaning
		E		Print an octal listing of the binary program in addition to the symbolic listing. This switch must be accompanied by the M switch.
		I		Translate the letter D in column 1 as a space, and treat the line as a normal FORTRAN statement.
		M		Include MACRO coding in output listing.
		N		Suppress output of error messages on the terminal.
	S			Produce code for execution on the KA10 if running on the KI10, and vice versa.
	T	T	T	Skip to the logical end of the magnetic tape.
	W	W	W	Rewind the magnetic tape.
	Z	Z		Zero the DECtape directory.
FORTRAN-10			C	Generate a file that can be input to the CREF program.
		E		Include the octal-formatted version of the object file in the listing.
			I	Translate the letter D in column 1 as a space, and treat the line as a normal FORTRAN statement.
	KA			Compile the program, producing code to execute on the KA10 processor.
	KI			Compile the program, producing code to execute on the KI10 processor.
		M		Add the mnemonic translation of the object code to the listing file.
		NOE	NOE	Suppress output of error messages to the terminal.
		NOW	NOW	Suppress output of warning messages to the terminal.

Table 1-2 (Cont.)
Processor Switches

Processor	Binary	Listing	Source	Meaning
	O			Perform optimization of global symbols when producing processor code.
			S	Perform compilation, checking for syntax errors only.
MACRO	A	A	A	Advance the magnetic tape reel by one file.
	B	B	B	Backspace the magnetic tape reel by one file.
		C		Produce 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 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.

1.5.8 Switches To The Linking Loaders

Two linking loaders are available to you on the DECsystem-10: the LOADER program and the LINK-10 program. (Refer to the LOADER Programmer's Reference Manual and the LINK-10 Programmer's Reference Manual for complete information.) In complex loading procedures, it may be necessary to pass switches to one of the linking loaders to direct its operation. The percent sign (%) character accomplishes this passing of switches.

1.5.8.1 **Passing LOADER Switches** - The % character takes an optional leading sign (+ or -) and one letter (or a sequence of digits and one letter) following it. To set a program origin of 6000 for file PROG3, you can type the following:

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

The %6000O switch, with the % character replaced with the / character, is passsed to the LOADER. The COMPIL program allows more than one LOADER switch to be specified:

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

(For a description of the /MAP switch, refer to the LOAD command description in Chapter 2.)

1.5.9 Passing LINK-10 Switches

The LINK-10 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-10 specification consists of the switch name and, optionally, a keyword and a value. The items in the specifications are separated by colons. To set a program origin of 6000 for file FILEB, you could type the following command:

```
.LOAD/LINK FILEA, %'SET:.LOW.:666'FILEB,FILEC
```

(For a description of the /LINK switch, refer to the LOAD command description in Chapter 2.)

CHAPTER 2

SYSTEMS COMMANDS AND PROGRAMS

Although there is one operating system for all configurations of the DECsystem-10, some commands may not be included in every DECsystem-10. Some installations delete certain features from their operating systems because of a constraint on core. Your system administrator can delete commands from the operating system by setting feature test switches when defining the system configuration via MONGEN.

In many cases, you can type one of two commands to run a program. For example, the indirect command MAKE and the direct command R TECO **both** run the TECO program. At some installations, the system administrator may have turned off the indirect command switch, but the switch implementing the direct command is always set. Therefore, you may always run a program with the R (or RUN) command, even if the switch implementing the corresponding indirect command has been turned off.

Although commands are arranged in alphabetical order in Chapter 2 for ease of reference, they can be divided into functional groups for ease of learning.

2.1 JOB INITIALIZATION COMMANDS

Because the system limits access to authorized persons, the following commands protect the system from unauthorized use.

INITIA	Performs standard system initialization for your terminal.
LOGIN	Gives you, the user, access to the system.

2.2 FACILITY ALLOCATION COMMANDS

The monitor allocates peripheral devices, file structure storage, and core memory to user jobs when they request these things. The monitor also protects these allocated facilities from interference by other users. The monitor incorporates software provisions to differentiate the central nodes from the remote nodes. Certain commands (e.g., ASSIGN, PLEASE) include node identification arguments to allow both user access and allocation of system resources at any node. This feature allows you considerable flexibility in allocating system facilities and directing input and output to a specific node.

By specifying a node number, you can assign devices and input data from a peripheral device at a node other than your own. In addition, by using the LOCATE command, you can logically establish your job at a node other than your physical node. If you do not include a node identification argument in your command line, the system automatically directs input and output to your logical node. Your logical node is the same as your physical node, if you have not issued the LOCATE command.

When the system assigns a nonsharable device to your job, the system removes that device from the monitor's pool of available resources. If another user attempts to reference or assign that device, the attempt fails. You should never leave the system without first returning your allocated facilities to the monitor pool. Allocated facilities are automatically returned to the monitor's pool of available resources when you deassign the resources or you kill your job. Until you return these facilities (resources), no other user can utilize them except through operator intervention.

Assignable devices (i.e., nondisk and nonspooled devices) in the monitor's pool of available resources are either unrestricted or restricted devices. Any user can assign an unrestricted device by using the ASSIGN command. Only privileged jobs (i.e., jobs logged in under [1,2] or running with the JACCT bit set) can assign restricted devices. However, nonprivileged users can have a restricted device assigned to them via the MOUNT command. The MOUNT command allows operator intervention for the selection or denial of a particular device; therefore, the operator controls the use of the assignable devices. This is particularly useful when there are multiprogramming batch and interactive jobs competing for the same devices and resources. The operator sets or removes the restricted status of a device via the OPSER commands :RESTRICT and :UNRESTRICT.

The facility allocation commands are listed below.

ASSIGN	Allocates an I/O device to your job without operator intervention.
CLOSE	Terminates I/O in progress on a specified device, but does not release the device.
CORE	Types or modifies the amount of core assigned to your job.
DEASSIGN	Returns devices assigned to your job to the monitor's pool of available devices.
DISMOUNT	Returns, via the operator, devices assigned to your job to the monitor's pool of available devices.
FINISH	Terminates I/O in progress on the specified device and performs the RELEASE monitor call or the DEASSIGN command.
LOCATE	Establishes your job at a specified node.
MOUNT	Allocates an I/O device to your job via the operator.
NODE	Associates a node name with a particular device.
REASSIGN	Gives the specified device to the designated job.
SET BLOCKSIZE	Sets the default block size for the specified magnetic tape unit.
SET CDR	Sets the file name for the next card reader spooling interrupt.

SET CPU Sets the CPU specification (i.e., CPxn, NO CPxn, ALL, ONLY CPxn) for your job.

SET DENSITY Sets the default density for the specified magnetic tape unit.

SET DSKPRI Sets the priority for your job's disk operations (data transfers and head positionings).

SET HOST Sets the central node for your job.

SET HPQ Sets the high priority scheduler run queue for your job.

SET PHYSIC Sets the maximum current physical page limit or guideline on a virtual memory system.

SET RETRY Enables or disables error retry on a specified magnetic tape unit.

SET SPOOL Adds devices to or deletes devices from the list of spooled devices for your job.

SET TTY Sets properties to be associated with your terminal.

SET VIRTUA Sets the current virtual page limit on a virtual memory system.

2.3 SOURCE FILE PREPARATION COMMANDS

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

CREATE Opens a new file on disk for creation with LINED.

EDIT Opens the specified file, already existing on disk, for editing with LINED.

MAKE Opens a new file on disk for creation with TECO.

TECO Opens a specified file that already exists for editing with TECO.

2.4 FILE MANIPULATION COMMANDS AND PROGRAMS

The commands and programs in this group allow you to manipulate files to any extent desired. You can list source files, DECtape directories, and disk directories on your terminal or the line printer, possibly via the spooling mechanism. You can delete or rename files from disk and DECtape. You can also transfer files between standard I/O devices, perform conversion between various core image formats, and read and write various directory files. You can allocate disk space for a new file or you can reallocate disk space for an existing file. Finally, you can place files in the system queues and obtain listings of entries in these queues.

ALCFIL Allocates space for a new file or reallocates space for an existing file.

BACKSPACE Spaces a magnetic tape backward the specified number of files or records.

BACKUP Saves and restores disk files.

COPY Transfers files from one device to another.

CPUNCH Places entries into the card punch output spooling queue.

DELETE Deletes files from DECtape or disk.

DIRECT Lists the directory entries for the specified arguments.

DTCOPY Copies the contents of one DECtape to another.

EOF Writes an end-of-file mark on the specified magnetic tape.

FAILSA Saves and restores disk files.

FILCOM Compares two versions of a file and outputs any differences.

FILE Provides remote control via the operator for DECtape-to-disk and disk-to-DECtape transfers.

GLOB Reads multiple binary files to produce an alphabetical cross-referenced listing of all global symbols encountered.

LABEL Writes an identifier on a DECtape.

LIST Writes the specified files on the line printer.

MTCOPY Copies entire contents of selected files from magnetic tapes.

PLOT Places entries into the plotter output spooling queue.

PRESERVE Renames the specified files with the standard protection inclusively ORed with 100.

PRINT Places entries into the line printer output spooling queue.

PROTECT Renames the specified files with the requested protections.

QUEUE Enters items into the specified system queue.

RENAME Changes one or more items of the file specification of files on DECtape or disk.

REWIND Rewinds a magnetic tape or a DECtape.

SET DEFAULT Sets the default protection of a file.

SKIP — Moves the specified magnetic tape forward the designated number of files or records or to the logical end of the tape.

SUBMIT — Places entries into the batch input queue.

TPUNCH — Places entires in the paper-tape punch output spooling queue.

TYPE — Types the specified files on your terminal.

UNLOAD — Rewinds and unloads the specified magnetic tape or DECtape.

ZERO — Clears the directory of the specified device.

2.5 OBJECT PROGRAM PREPARATION COMMANDS AND PROGRAMS

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

COMPILE — Produces relocatable binary files (.REL extensions) for the specified source files.

CREF — Lists on the line printer any cross-referenced listing files generated by a previous COMPILE, LOAD, EXECUTE, or DEBUG command.

DEBUG — Produces relocatable binary files for the specified source files, loads the .REL files, and prepares for debugging with DDT.

EXECUTE — Produces relocatable binary files for the specified source files, loads the .REL files, and begins execution.

FUDGE — Creates a library .REL file by reading a temporary file generated by a previous COMPILE, LOAD, EXECUTE, or DEBUG command.

LOAD — Produces relocatable binary files for the specified files and loads the .REL files generated.

MAKLIB — Updates the files containing relocatable binary programs and manipulates programs within the program files.

OSAVE — Writes a core image of your core area on the specified device.

NSAVE — Writes a core image of your core area on the specified device.

OSSAVE — Writes a core image of your core area on the specified device. When loaded with GET (or RUN) the high segment will be sharable.

NSSAVE — Writes a core image of your core area on the specified device. When loaded with GET (or RUN) the high segment will be sharable.

2.6 OBJECT PROGRAM CONTROL COMMANDS

By using commands in this group, you can load core image files from retrievable storage devices (ie., disk, DECtape, magnetic tape). These files can be retrieved and controlled from your terminal. You address files stored on disk and DECtape by file name. To reference a file on a magnetic tape, you must pre-position the tape to the beginning of the file. (Refer to DECsystem-10 Monitor Calls for a description of the job data area location referenced by the command descriptions in this group of commands.)

CCONT Continues the program from the point at which you interrupted it, but leaves your terminal in monitor mode.

CONT Continues the program from the point at which you interrupted it.

CSTART Begins the execution of a program that you loaded either with a GET command or that you interrupted, but leaves your terminal in monitor mode.

DDT Copies the saved program counter and then starts the program at the beginning address of DDT.

GET Loads a core image from the specified device, but does not begin execution.

HALT Stops your job and stores the program counter in the job data area.

JCONT Continues the specified job, if it was waiting for operator intervention.

R Loads a core image from the system device (:SYS) and starts it at the location specified in the file.

REENTER Starts the program at an alternate entry point specified by the program.

RUN Loads a core image from the specified device and starts it at the location specified in the file.

SET BREAK Halts a program at a specified location on various conditions.

SET DSKFUL Controls your job when you have exhausted your assigned disk space.

START Begins execution of a program that you either previously loaded with the GET command or interrupted.

2.7 OBJECT PROGRAM EXAMINATION COMMANDS

The commands in this group help you to examine and analyze an object program. Dumps of your core area can be taken, and later processed by the system program DUMP, according to the arguments you specify.

D(eposit) Deposits information in your core area.

DCORE Writes a core image file in your core area.

DUMP	Writes a core image file and analyzes the written file to provide printable output.
E(xamine)	Examines the specified core location in your area.

2.8 MULTIPLE JOB CONTROL COMMANDS AND PROGRAM

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

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

2.9 JOB TERMINATION COMMANDS

When you leave the system, all facilities allocated to your job must be returned to the monitor's pool of available resources so that they will be available for other users.

KJOB	Gives up access to the system.

2.10 SENDING MESSAGES

The commands in this group allow you to communicate with other users of the system or with operators at any node. In addition, you may record information in a disk file to be read by the operations staff at your installation at a later time.

GRIPE	Accepts text from you and records it in a disk file for the operations staff at your installation.
PLEASE	Provides two-way communication between you and the operator.
SEND	Provides one-way communication.

2.11 JOB INFORMATION COMMANDS AND PROGRAMS

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

DSK Types disk usage for the combined structures of the specified job.

PJOB Outputs the job number to which your terminal is currently attached.

QUOLST Types the number of blocks used, the logged-in quota, and the logged-out quota for each file structure to which you have access, followed by the number of free blocks left on that structure.

SETSRC Manipulates your job search list or the system's search list.

SET TIME Sets the central processor time limit for your job.

SET WATCH Sets the output of incremental job statistics.

TIME Outputs the running time for the specified job.

USESTAT Outputs status information pertaining to your current job.

2.12 SYSTEM INFORMATION COMMANDS AND PROGRAMS

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

DAYTIME Types the current date, followed by the time of day.

RESOURCES Outputs the names of all available devices (except TTYs and PTYs), all file structures, and all physical units not in file structures.

SCHED Outputs the schedule bits set by the operator.

SYSTAT Prints system information about the current system.

VERSION Outputs the version number of a program.

WHERE Outputs the node number at which the specified device is located.

Function

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

Command Format

```
R ALCFIL
```

The ALCFIL program responds with

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

You may respond with one of the below.

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

where dev: is a file structure name or a physical unit name. If you omit dev:, the system assumes DSK. If you omit one of the other arguments, the system assumes 0. If you specify a file name, the system allocates the number of blocks currently allocated. ALCFIL responds with the following.

```
ALLOCATE?
```

You may type n or n,m (decimal numbers)

n = total number of blocks to be allocated for the file.

m = logical block within the file structure or unit (depending on the device you specified) where the system begins allocation.

If the system cannot allocate the total number of blocks that you requested, the system gives you a partial allocation and prints the following message:

```
PARTIAL ALLOCATION ONLY
```

You can issue the DIRECT command with the ALLOCATE switch to determine the number of blocks that the system allocated. If the system can allocate all of the blocks that you requested, it prints the following message:

```
ALLOCATED
```

Because the system executes an extended ENTER to allocate the new blocks, the file does not have to exist before the system can allocate the blocks. (Refer to DECsystem-10 Monitor Calls for a description of the extended ENTER monitor call.)

ALCFIL Program (Cont.)

Characteristics

The R ALCFIL command:

Places your terminal in user mode.

Runs the ALCFIL program, destroying your core image.

Example

```
.R ALCFIL


/H FOR HELP
FILE? TEST.ALG
ALLOCATE? 17

ALLOCATED
FILE? PROG1.FOR
ALLOCATE? 10

ALLOCATED
FILE? PROG1.REL
ALLOCATE? 1000

PARTIAL ALLOCATION ONLY
FILE? ^C
```

ASSIGN Command

Function

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

Command Formats

$$\text{ASSIGN} \left\{\begin{matrix}\text{devu}\\ \text{devnuu}\\ \text{dev}\\ \text{devnn}\end{matrix}\right\} \quad \text{logical-device}$$

devu is any physical device listed in Table 1-1, followed by a unit number or any file structure name. The monitor attempts to ASSIGN the device number you specifically requested at your current node. (Example: LPT6).

devnuu is any physical device listed in Table 1-1, followed by a 2-digit node number, followed by a unit number. The monitor attempts to ASSIGN the device you specified at the node you specified. (Example: LPT226).

dev is any physical device listed in Table 1-1. The monitor attempts to ASSIGN a device of the type you specified at your job's logical node. (Example: LPT).

devnn is any physical device listed in Table 1-1, followed by a 2-digit node number. The monitor attempts to ASSIGN a device of the type you specified at the node number you specified.

logical-device is a logical device name that you request be assigned to the specified physical device. Note that this argument is optional. Except for disk devices, you can assign only one logical name to a device. Subsequent ASSIGN commands to all devices except disk devices replace the old logical name with the new one. Logical names are disassociated from all devices when you issue the DEASSIGN command.

Characteristics

The ASSIGN command:

Leaves your terminal in monitor mode.

Restrictions

You cannot use a comma to separate the logical and physical device names. If you do use a comma, the monitor terminates its scan at the comma; therefore, the monitor does not assign a logical name.

Nonprivileged jobs (i.e., jobs not logged in as [1,2] or running with JACCT set) can use this command only to allocate unrestricted I/O devices. Nonprivileged jobs can obtain restricted devices by issuing the MOUNT command. A privileged job can allocate both restricted and unrestricted devices via the ASSIGN command.

ASSIGN Command (Cont.)

Examples

```
.ASSIGN LPT2:↵
LPT262 assigned

.ASSIGN LPT:↵
?Assigned to Job 3,1

.ASSIGN DTA2:↵
DTA262 assigned

.ASSIGN LPT226↵
?No such device

.ASSIGN DTA010↵
?No such device
```

ATTACH Command

Function

The ATTACH command detaches your current job and connects your terminal to a detached job. If you are an operator, you can attach your job to a device (Refer to Appendix A of DECsystem-10 Operator's Guide.)

Command Format

1. ATTACH job [project,programmer]

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

    [project,programmer] is the project-programmer number of the originator of the desired job. You may omit this argument if the project-programmer number is the same as the one for your job. The operator (device OPR) or anyone logged in under [1,2] may always attach to a job although another terminal is attached, provided that you type the proper project-programmer number.

    To prevent you from attaching to jobs without knowing the password associated with the job, the system temporarily creates a new job when you specify the project-programmer number argument. This temporary job runs LOGIN to check the password. This action can result in your job being unable to attach to the specified job if the system's job capacity is exceeded by the creation of the temporary job. However, your job is still detached even though there are no available jobs. The operator or anyone logged in as [1,2] can always attach to another job, because privileged jobs do not cause the system to create a temporary job.

2. ATTACH dev:

    This format is operator-priviliged. (Refer to the DECsystem-10 Operator's Guide.)

Characteristics

The ATTACH command:

Leaves your terminal in monitor mode.
Does not require LOGIN.

Restrictions

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

Examples

```
.ATTACH 1↵                          You attach to job 1 from job
                                    5.
```

ATTACH Command (Cont.)

```
FROM JOB 5
```

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

```
.LOGIN 27,235↵
```

You log in to the system.

```
JOB   7 R5743A SYS #40/2 TTY 116
```

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

```
PASSWORD:__________↵
103417-JUNE-1977THUR

.ATTACH 35[50,27]↵
FROM JOB 7
PASSWORD:__________↵
```

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

```
.ATTACH 7↵
?CAN'T ATT TO JOB
```

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

```
.K/F↵
JOB 7, USER [50,27] LOGGED OFF TTY116 L635 17-JUN-77
RUNTIME 6.58 SEC.
```

You killed job number 35. Your terminal is currently not attached to any job.

```
.ATTACH 7↵
?CAN'T ATT TO JOB
```

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

```
.ATTACH 7[27,235]↵
PASSWORD:
```

The system accepts the command and requests from you the password. The system does not print the message FROM JOBn because the terminal was not attached to any job. Then, the system attaches your terminal to job 7.

BACKSPACE Command

Function

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

```
MTAn: (M#nB)=
MTAn: (M#nP)=
```

SAVed files on magnetic tape always contain two files: a high segment file and a low-segment file. If one of the segments is missing, the system writes a null file containing one record. The system writes the file before the EOF for the missing segment. To backspace over a SAVed file, you must backspace two files.

Command Formats

1. BACKSPACE MTAn: x FILES

   skips backward x files.

2. BACKSPACE MTAn: x RECORDS

   skips backward x records.

Characteristics

The BACKSPACE command:

Leaves your terminal in monitor mode.
Runs the PIP program, destroying your core image.

Examples

```
.BACKSPACE MTA012:7 RECORDS↵
```

Backspaces 7 records on the tape on MTA0 at node 12.

```
.BACKSPACE MTA013:11 FILES↵
```

Backspaces 11 files on the tape on MTA0 at node 13.

BACKUP Program

Function

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

Command Format

```
.R BACKUP
*/switch
```

BACKUP switches are described in the BACKUP Specification and the Monitor Installation Guide.

Characteristics

The R BACKUP command:

Places your terminal in user mode.
Runs the BACKUP program destroying your core image.

Example

```
.R BACKUP↵
/TAPE↵
/REWIND↵
/RESTORE↵
/UNLOAD↵
.
.
.
```

COMPILE Command

Function

The COMPILE command produces relocatable binary files (.REL files) and/or compilation listings for specified source program files. The system determines the proper language translator to use by the source file extension or by switches you specified in the command string. If you did not supply any switches in the command string, the system uses the following translators:

Source File Extension	Translator Used
.ALG	ALGOL compiler
.BLI, .B10	BLIS10 compiler
.CBL	COBOL compiler
.F4 or .FOR	FORTRAN compiler
.FAI	FAIL compiler
.FCL	FOCAL compiler
.LSP	LISP compiler
.MAC	MACRO assembler
.P11	MACY11 assembler
.SAI	SAIL compiler
.SNO	SNOBOL compiler
	Standard translator, which is usually FORTRAN; refer to Section 1.5.6.

NOTE

The translator cannot be changed with a switch but it can be specified with a switch when the source file has a nonrecognizable or null translator extension. For a source file with a recognizable translator extension, refer to the above list.

Normally, the system translates the source file if there is no corresponding binary (.REL) file, or if the source file's date and time is equal to or later than the binary file's date and time. If the binary file is newer than the source file, the system does not translate the source file, and uses the current .REL file. You can use switches to override this action.

NOTE

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

BLIS10, FAIL, FOCAL, LISP, MACY11, SAIL, and SNOBOL will be recognized as translators only if the appropriate assembly switches are set.

The extensions .F4 and .FOR indicate the file is a FORTRAN language file; they do not represent the particular FORTRAN compiler (either F4 or FORTRAN-10) to be used in the translation. The controller used is controlled by the switches /F40 and /F10.

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 core if the monitor has the TMPCOR features (defined via MONGEN). When you issue one of these commands without an argument, it causes the system to use the arguments saved in the temporary file. (Refer to Section 1.5.)

The COMPIL command can accept a range of four command constructions: @ (indirect commands), +, =, and < >. (Section 1.5 describes each construction.)

Command Format

COMPILE list

list 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. (Refer to Section 1.4.2.4.)

You may use the following switches as temporary or permanent command string modifiers. (Refer to Section 1.5.5.) Only switches relevant to COMPIL follow; all switches allowed with LOAD, EXECUTE and DEBUG may be used with COMPILE; all others are intentionally omitted.

Switch	File Control Function
/ALGOL	Compile with ALGOL. (Assumed wherever .ALG extension is used.)
/BIN	Generate a binary file for each file compiled. The file name of the binary file follows the standard convention for determining the file name of the output file. (Refer to Sections 1.5.2 and 1.5.3.) The extension is .REL. This is the default action.
/BLISS	Compile the file with BLISS10. Assumed for files with the extension of .B10 and .BLI. BLISS10 is recognized as a translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.
/COBOL	Compile the file with COBOL. Assumed for files with the extension of .CBL.
/COMPILE	Force a compilation on this file even if a binary file exists with a newer date and time than the source file. This switch is used to obtain an extra compilation (e.g., to obtain a listing of the compilation) because, normally, compilation is not performed if the binary file is newer than the source file.

Switch — File Control Function

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

/FORTRAN — Compile the file with a FORTRAN compiler. Assumed for files with either the extension of .F4 or .FOR and for all files with nonrecognizable translator extensions, if FORTRAN is the standard translator. This switch is needed if the file has a nonrecognizable translator extension and FORTRAN is not the standard translator or is not the current default (e.g., COMPILE/ALGOL FILEA, FILEB, FILC/FORTRAN).

/FUDGE — Create a disk file containing the names of the .REL files produced by the command string. When the FUDGE command is given, PIP reads this file to generate a library .REL file. (Refer to the <u>MAKLIB User's Guide</u>.) Arguments to this switch are:

```
/FUDGE:DEV:FILE.EXT[PROJ,PROG]
```

dev: is the device on which to write the file. If you do not specify a device, the system assumes DSK.

file.ext is the name of the library file. You must supply a file name but 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 one.

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

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

COMPILE Command (Cont.)

Switch — File Control Function

/K?10 — Designate the machine on which the program will execute once it has been loaded. These switches are needed for FORTRAN-10 programs because the compiler generates different codes for the KA10 and KI10 processors. The default is the processor on the computer executing the command. The ? can be A, I, or L.

/LIST — Generate a disk listing file for each file compiled. The file name of the listing file follows the standard conventions for determining the file name of the output file. (Refer to Sections 1.5.2 and 1.5.3.) The extension These files can be listed later with the LIST command. If the line printer is being spooled for your job, the listing files are written on device LPT and are automatically spooled at LOGOUT time. Unless you specify this switch, listing files are not generated.

/MACRO — Assemble the file with MACRO. Assumed for files with the extension of .MAC.

/MACY11 — Assemble the file with MACY11. Assumed for files with the extension .P11. Note that MACY11 is recognized as a translator only if the appropriate assembly switch is set. However, this assembly switch is not supported.

/MANTIS — Compile the program with the MANTIS debugging information. This switch affects F40 files only. MANTIS is recognized as the debugging program only if the appropriate assembly switch is set. However, the switch setting is not supported.

/NEW — Run the appropriate language translator from the experimental library (device NEW:) area [1,5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:.

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

/NOBIN — Do not generate binary files. Unless this switch is given, binary files are generated. This switch, when combined with the /CREF or /LIST switch, is useful when compiling programs solely for the purpose of generating listings.

Switch — File Control Function

/NOCOMPILE — Complement of the /COMPILE switch, which does not force a compilation of a source file whose date is not as recent as the date on the binary file. /NOCOMPILE is the default action.

/NOLIST — Do not generate listing files. This is the default action.

/NOMANTIS — Compile the program without the MANTIS debugging information. This switch affects F40 files only.

/OLD — RUN the appropriate language translator from the system library of old programs (device OLD:), which resides on the [1,3] disk area. If the translator does not exist on device OLD:, try to obtain it from device SYS:.

/SELF — Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for an individual who keeps a private copy of a translator in his own disk area. System programmers occasionally keep experimental versions of standard translators in their disk areas to test new features.

/SNOBOL — Compile the file with SNOBOL. Assumed for files with the extension .SNO. Note that SNOBOL is recognized as a translator only if the appropriate assembly switch is set. However, this assembly switch is not supported.

NOTE

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

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

However, the following is valid:

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

Characteristics

The COMPIL command:

Leaves your terminal in monitor mode.

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

Restrictions

The wildcard construction cannot be used.

COMPILE Command (Cont.)

Examples

```
.COMPILE PROG,TEST.AC,MANAGE/COBOL↵
```

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

```
.COMPILE /LIST SIGN.MAC, TABLES/NOLIST,MULTI.ALG↵
```

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

```
.COMPILE/CREF/COMPILE DIVIDE,SUBTRC,ADD↵
```

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

```
.COMPILE /FUDGE:MONITOR.REL@LIBALL↵
```

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

```
.COMPILE OUTPUT=MTA010:(W,S,M)/L↵
```

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

CONTINUE Command

Function

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

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

Command Format

CONTINUE

Characteristics

The CONTINUE command:

Places your terminal in user mode.
Requires core.
Does not require LOGIN.

Example

```
.RUN LOOP↵     Run a program called LOOP in your disk area.

^C             Stop the program.
                          ^C

.DAYTIME↵      Check the time of day and the date.

17-JUN-77  11:03:21

.CONT↵         Continue the program.
```

COPY Command

Function

The COPY command transfers files from one standard device to another. The command string can contain one device output specification and any number of input specifications. You use the equal sign (=) to separate the destination (output) side from the source (input) side. This command runs PIP and performs the basic PIP function of transferring files. To copy DECtapes or magnetic tapes, refer to the command descriptions of DTCOPY and MTCOPY.

Command Format

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

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 assumes the input file name. If you transfer many input files to one output file, the system combines the files. If you transfer many input files to the same number of output files, the system keeps the files as separate files. You can use the wildcard construction with the COPY command.

[directory] is the disk area in which either 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 disk area is assumed. You may transfer files to or from another disk area only if you have access to that area.

<nnn> is the protection code to be given to the output file(s). 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.

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

Characteristics

The COPY command:

Leaves your terminal in monitor mode.
Runs the PIP program, destroying your core image.

Examples

```
.COPY YOURS.CBL[20,17]=MINE.CBL↵
```

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

Function

The CORE command types out or modifies the amount of core assigned to your job. Because programs usually allocate core, you generally do not need this command. It is included for completeness and is used more frequently in nonswapping systems than in swapping systems.

If your job is locked in core and you issue this command with a nonzero argument, it cannot be satisfied. Therefore, the system gives you an erroneous return.

Command Format

CORE n

n is a decimal number; this argument is optional.

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

If n is 0, the low and high segments disappear 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 high plus the minimum low segment size, n plus the high segment is assumed.

Core arguments can be specified in units of 1024 words or 512-word pages when you follow n with the letter K or P, respectively. For example, 3P represents three pages or 1536 words. If you do not specify K or P, K is assumed (i.e., 1024 words).

On systems with the KA10 processor, 1024 words is the minimum unit of allocation; therefore, all arguments are rounded up to the nearest multiple of 1024 words. For example, 3P on the KA10 is treated the same as 2K.

Characteristics

The CORE command:

Leaves your terminal in monitor mode.
Does not operate when a device is currently transmitting data.

Examples

```
.CORE 135P↵

.CORE 556P↵
?Virt. mem. assigned 136P (Current limit: 512P Max limit: 512P)
Phys. mem. assigned 136P (Guideline: 512P Max limit: 139P)
Swap space left: 1118P

.CORE↵
Virt. mem. assigned 136P (Current limit: 512P Max limit: 512P)
Phys. mem. assigned 136P (Guideline: 512P Max limit: 139P)
Swap space left: 1120P
```

CPUNCH Command

Function

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

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

Command Format

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

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

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

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

file.ext is a list of the names of the files. The file name is optional. If you do not specify a file name, the default for the first file name is *, the default for the subsequent files is the last file name used. You can omit the extension; its default is .CDP.

[directory] is a directory to which you have access. If you do not specify a directory; your directory area is assumed.

You can obtain the listing of entries in the card punch output queue for a specific project-programmer number by following the command with the desired project-programmer number enclosed in square brackets (e.g., CPUNCH [40,200]). If all arguments to the command are omitted (only the command name is specified), the listing of all entries in the card punch queue for all jobs of all users is output.

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

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

2. General - Each switch in this category can appear only once in the command string because the category affects the entire request. The switch used can appear anywhere in the command string.

3. File control - Any number of these switches can appear in the command string because this category is specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the file name, it becomes the default for subsequent files. For example, the command string

   ```
   CPUNCH FILEA, FILEB/DISP:REN,FILC
   ```

   indicates that the DISPOSE switch is only for FILEB. The command string:

   ```
   CPUNCH/DISP:REN FILEA,FILEB,FILEC
   ```

   indicates that the DISPOSE switch applies to all three files.

Note that if you omit the argument to a switch, you must also omit the colon preceding the argument. Otherwise, the argument is assumed to be zero and not the default value. (Refer to the QUEUE command description for a complete list of the switches and explanations of their functions.)

Characteristics

The CPUNCH command:

Leaves your terminal in monitor mode.
Runs the QUEUE program, destroying your core area.
Does not require LOGIN when only queue listings are desired.

Examples

```
.CPUNCH SYSTAT.MAC/PUNCH:ASCII↵
```

Punch the file SYSTAT.MAC in ASCII format.

```
.CPUNCH SYSTAT.REL/PUNCH:BINARY/AFTER:17:00↵
```

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

CREATE Command

Function

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

Command Format

CREATE file.ext

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

Characteristics

The CREATE command:

Places your terminal in user mode.
Runs the LINED program, destroying your core image.

Examples

```
.CREATE TEST.FOR↵

*
```

Function

The CREDIR program is used to create directories and sub-directories. It will construct sub-file directories (SFDs) on all structures selected, including any higher level directories that may be necessary.

When CREDIR is running on [1,2], it creates UFDs on any selected structure or set of structures. You specify the quotas to be associated with these UFDs by using appropriate switches or by answering questions during the CREDIR dialogue.

CREDIR can be used for creating ersatz device directories. It allows you to create directories on all structures associated with a specified ersatz device. As each directory is created, CREDIR informs you of the name of the unit and the protection code associated with the directory.

Command Format

R CREDIR

CREDIR types out CREATE DIRECTORY:, after which you respond with the device name and the directory specification. you can specify an ersatz device name and you can supply switches in the argument. Device DSK: is assumed if you do not specify a device. If you do not specify a protection code for the directory, the system gives the directory a default protection code of 775.

Example

```
.R CREDIR↵

CREATE DIRECTORY: DSKB:[,,A,B,C]↵
  CREATED DSKB:[27,4072,A].SFD/PROTECTION:775
  CREATED DSKB:[27,4072,A,B].SFD/PROTECTION:775
  CREATED DSKB:[27,4072,A,B,C].SFD/PROTECTION:775
CREATE DIRECTORY:^C
```

CREF Command

Function

The CREF command runs the CREF program. In doing so, it lists on the line printer any cross-referenced listing files that were generated since your job was initiated. You generate cross-referenced files by using the /CREF switch in any previous COMPILE, LOAD, EXECUTE or DEBUG commands. The file containing the names of these CREF-listing files is then deleted so that subsequent CREF commands will not list them again. The output goes either to LPT immediately or to the disk to be spooled later to LPT. When the logical device name LPT is assigned to a device other than the line printer, the CREF files are stored on that device with the same file name and the extension .LST. (Refer to the CREF manual in the DECsystem-10 Software Notebooks for more information.)

Command Format

CREF and CREF file-specification

where file-specification is device:filename.extension[ppn]. when you supply a file specification in the command line, cref produces a cross0referenced listing file for the specfied file.

Characteristics

The CREF command:

Leaves your terminal in monitor mode.

Runs the CREF program, destroying your core image.

Example

`.COMPILE /CREF@PROMAC` ↵	Compile the files contained in the command file PROMAC and produce CREF input compatible cross-referenced listing files on the disk.
`.CREF` ↵	Process and list the cross-referenced listing files produced by the COMPILE command.
`.LOAD/C/MAP:NAME @CONALL` ↵	Compile and load the files contained in the command file CONALL. Produce a loader map with the file name NAME and CREF input compatible cross-referenced listing files on the disk.
`.ASSIGN MTA1 LPT` ↵ `ASSIGNED TO JOB`	Assign the logical name LPT to MTA1.
`.CREF` ↵	Store the CREF files on MTA1 to be listed at a later time.

CSTART Command
CCONTINUE Command

Function

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

Command Format

```
CSTART addr
CCONTINUE
```

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

To use:

1. Begin the program with your terminal in user mode.
2. Type control information to the program, then Type a CTRL/C to halt your job with your terminal in monitor mode.
3. Type CCONTINUE to allow your job to continue running and leave your terminal in monitor mode.
4. You can now enter additional commands from your terminal.

Characteristics

The CSTART and CCONTINUE commands:

Leave your terminal in monitor mode.
Require core.

Restrictions

The commands should not be used when your program (which continues to run) is also requesting input from the terminal. These commands are not available to batch users.

CSTART Command (Cont.)
CCONTINUE Command

Example

```
.TYPE TEST.ALG↵
BEGIN INTEGER S,C,B,A
!THIS PROGRAM WILL ADD THREE NUMBERS AND;
!PRINT THE RESULT:
WRITE ("[2C] WHAT ARE THE NUMBERS TO ADD [1C] ");
READ (S);
READ (C);
READ (B);
A:=C+S+D;
WRITE ("[2C] 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 ("[2C] ");
END
.EXECUTE TEST.ALG↵
ALGOL:TEST
LINK: LOADING
[LNKXCT TEST EXECUTION]

 WHAT ARE THE NUMBERS TO ADD
23    45↵

^C

.CCONT↵

.TIME↵
7.50
7.50
KILO-CORE-SEC=64

.SYSTAT↵
?PLEASE TYPE CONTROL-C FIRST

.TIME↵
0.00
7.50
KILO-CORE-SEC=64

.^C
```

D(eposit) Command

Function

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

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

Command Format

D lh rh addr

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 desposited 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 addr, the system deposits the data into the location following the last D addr command or in the location of the last E addr command (whichever was last).

Characteristics

The D command:

Leaves your terminal in monitor mode.
Requires core.

Example

```
.D 266000 2616 141↵                      Deposit in location 141.

.E 140↵                                  Examine location 140

000140/ 265720 40011 .D 47000 1↵         Because addr is omitted,
                                         the deposit is in the
                                         location of the last E
                                         command.

.E↵
000140/ 047000 000001  .                 Examine is location
                                         specified in the previous
                                         D command.
```

DAYTIME Command

Function

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

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

where:   dd = day
         mmm = month
         yy = year
         hh = hours
         mm = minutes
         ss = seconds to the nearest hundredth
```

Command Format

DAYTIME

Characteristics

The DAYTIME command

Leaves your terminal in monitor mode.
Does not require LOGIN.
Does not destroy your core area.

Examples

```
.DAYTIME↵
28-Jul-77 15:08:12

.DA↵
28-Jul-77 15:08:19

.
```

Function

The DCORE command causes the DAEMON program to write a core-image file of your core area that includes all accumulators and all relevant job tables. The job can contine to run; the DCORE command does not destroy your core image. The file produced may be processed later by the DUMP program, if you wish.

The DAEMON-written file consists of five categories: JOB, CONFIGURATION, DDB, CORE, and FEATURES. Each category has a 2-word header, the first word contains the category number and the second word contains the number of data words in the category. The categories are listed below.

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

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

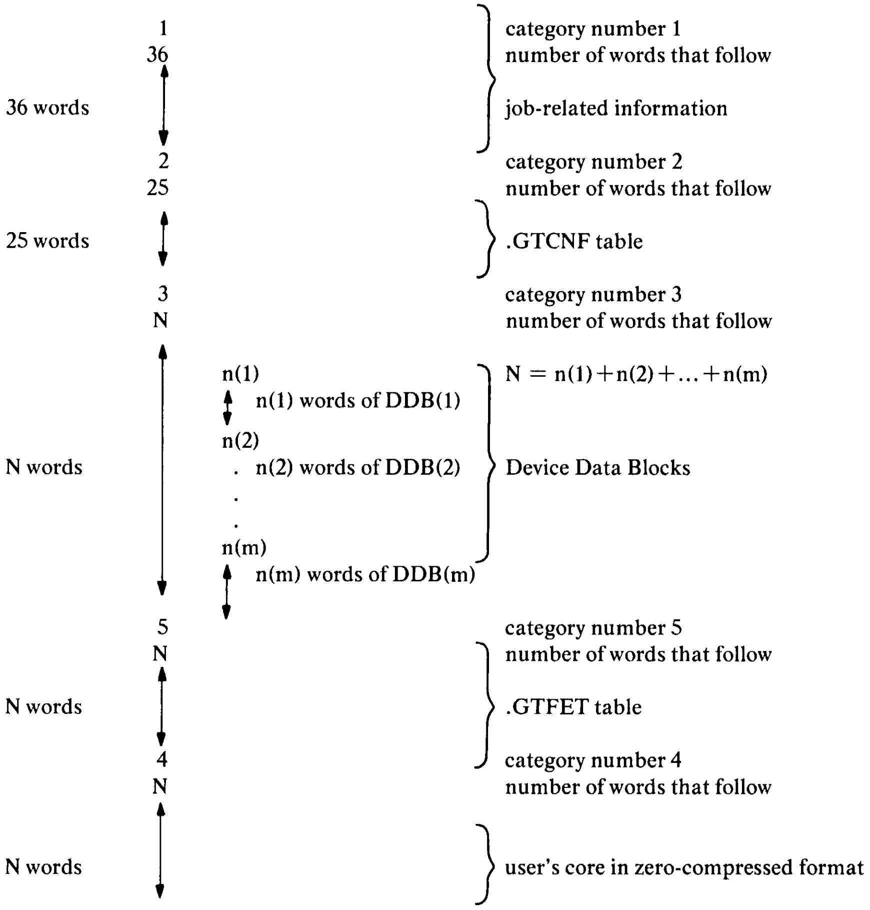


Figure 2-1 DAEMON-written File

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

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

Category 2 contains the entries in the .GTCNF GETTAB table. (Refer to the UUOSYM listing in DECsystem-10 Monitor Calls.)

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

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

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

Command Format

DCORE dev:name.ext[directory]

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

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

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

Characteristics

The DCORE command:

Leaves your terminal in monitor mode.
Runs the DAEMON program, destroying your core area.
Can be continued after command.

Examples

.DCORE↵	The core image is written on your area of the disk with the name nnnDAE.TMP, where nnn is your job number.
.DCORE DSKB:FILEC↵	The core image file is written in your area on DSKB with the name FILEC.DAE.

DDT Command

Function

The DDT command copies the saved program counter value from .JBPC into .JBOPC and starts the program at an alternate entry point specified in .JBDDT (beginning address of DDT as set by LINK-10). DDT contains command to allow you to start or resume at any desired address. (Refer to DECsystem-10 Monitor Calls for a description of the above mentioned Job Data Area locations.)

If your job was executing a monitor call when interrupted (in exec mode and not in TTY input wait or SLEEP mode), the monitor sets a status bit (UTRP) and continues the job at the location at which it was interrupted. When the 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 in exec mode and in TTY INPUT WAIT or SLEEP MODE, the trap to the DDT address occurs immediately and .JBOPC contains the address of the monitor call. If your job is in user mode, the trap also occurs immediately. Therefore, it is always possible to continue the interrupted program after trapping to DDT by executing a JRSTF @ .JBOPC.

(For addtional information on the DDT program, refer to DDT Programmer's Reference Manual.)

Command Format

DDT

Characteristics

The DDT command:

Places your terminal in user mode.
Requires core.
Requires that you have a job number.
If the system is configured with the virtual memory option and .JBDDT is zero, the system will read in SYS:DDT.VMX starting at location 700000.

Examples

```
.TYPE LOOP.MAC↲              Type an undebugged program.
LOOP:JRST LOOP
END LOOP


^Z                           Save the program.
.OSAVE↲
JOB SAVED


.DDT↲                        Enter DDT.

LOOP/JRST LOOP CALLI 12      Fix the program.
     JRSTF @.JBOPC@X

EXIT
.
```

DEASSIGN Command

Function

The DEASSIGN command returns one or more devices currently ASSIGNed to your job back to the monitor's pool of available devices, and clears any logical names. Restricted devices are returned to the restricted pool, and unrestricted devices are returned to the unrestricted pool. An INITed device is not returned to the monitor's pool unless you have issued a RELEASE monitor call, only the logical name is cleared. This command is provided for programs that are not terminating or programs that are being debugged. The command, applied to DECtapes, clears the copy of the directory currently in core, forcing the next directory reference to read a new copy form the tape . (Refer to DECsystem-10 Monitor Calls for further details.)

Command Format

DEASSIGN dev

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 the logical name associated with the controlling terminal.

Characteristics

The DEASSIGN command:

Leaves your terminal in monitor mode.

Examples

.DEASSIGN LPT↵	The line printer is returned to the monitor's pool of available resources.
.DEASSIGN↵	All devices assigned to your job are returned to the monitor's pool of available devices.

DEBUG Command

Function

The DEBUG command translates the specified source files, if necessary (function of the COMPILE command), loads the REL files generated (function of the LOAD command), and prepares for debugging. A system debugging program is loaded first, followed by your program with local symbols. Upon completion of loading, control is transferred to the debugging program. The debugging program 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 first file is a file other than a COBOL SOURCE file, DDT (the Dynamic Debugging Technique) is loaded.

NOTE

> This command runs the COMPIL program, which interprets the command before running the appropriate language translator, the LINK-10, and the appropriate debugger.

When the first file is a previously translated program (i.e., with extension .REL) and COBDDT is desired as the debugging program, the /COBOL switch must appear in the command string. For example:

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

(Refer to DDT documentation for a description of DDT commands, and refer to COBOL documentation for a description of COBDDT.)

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

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

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

Command Format

DEBUG list

> list is a single file specification or a string of file specifications, separated by commas. A file specifications consists of a device name, a file name (with or without an extension), and a directory name. Refer to Section 1.4.2.4.

The following switches can be used to modify to command string. These switches are temporary or permanent unless stated otherwise.

Switch	Meaning
/ALGOL	Compile the file with ALGOL. Assumed for files with the extension of .ALG.
/BIN	Generate a binary file for each file compiled. The file name of the binary file follows the standard conventions for determining the file name of the output file; refer to Sections 1.5.2 and 1.5.3. The extension is .REL. This is the default action.
/BLISS	Compile the file with BLIS10. Assumed for files with the extension of .B10 and .BLI.

NOTE

BLIS10 is recognized as a translator only if the appropriate switch is set. However, this assembly switch setting is not supported.

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

DEBUG Command (Cont.)

Switch — Meaning

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

/FUDGE:dev:file.ext[project,programmer]

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

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

[project,programmer] is the directory area in which the system is to place the file. If you do not supply this argument, the system assumes your directory area.

This switch is a permanent switch so far as it pertains to all REL files generated by the command string.

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

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

/K?10 — Designate the machine on which the program will execute once it has been loaded. These switches are needed for FORTRAN-10 programs because the compiler generates different codes for the processors. The default is the processor on which your program is running. The ? can be A, I, or L.

/LIBRARY — Load the files 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. (Refer to LINK-10 documentation.)

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

Switch Meaning

/LIST Generate a disk listing file for each file compiled. The file name for the listing file follows the standard conventions for determining the name of the output file. (Refer to Sections 1.5.2 and 1.5.3.) The extension is .LST. These files can be listed later with the LIST command. If the line printer is being spooled for this job, the listing files are written on device LPT and are automatically spooled at LOGOUT time. Unless this switch is specified, listing files are not generated.

/LMAP Produce a loader map during the loading process (same action as /MAP) containing the local symbols.

/LOADER Cause the file to be loaded by the LOADER program.

/MACRO Assemble the file with MACRO. Assumed for files with the .MAC extension.

/MACY11 Assemble the file with MACY11. Assumed for files with the .P11 extension.

/MANTIS Compile the file with the MANTIS debugging information. This switch affects F40 programs only.

/MAP Produce a loader map during the loading process. When this switch is encountered, the system requests a loader map from the loader. After the library search of the system libraries, the system writes the map in your disk area with either the file name you specified or the default file name MAP.MAP, if loading is performed by LOADER, or nnnLNK.MAP, if performed by LINK-10. This switch is an exception to the permanent switch rule because it causes only one map to be produced, even though it may appear as a permanent switch.

/NEW Run the appropriate language translator from the experimental system library (device NEW:) area [1,5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:.

/NOBIN Do not generate binary files. Unless this switch is given, binary files are generated. This switch, when combined with the /LIST or /CREF switch is useful when compiling programs solely for the purpose of generating listing.

/NOCOMPILE Complement of the /COMPILE switch by not forcing a compilation on a source file whose date is not as recent as the date on the binary file. Note that this switch is not the same as the /REL switch, which turns off all compilation, even if the source file is newer than the REL file. /NOCOMPILE is the default action.

/NOLIST Do not generate listing files. This is the default action.

DEBUG Command (Cont.)

Switch — Meaning

/NOMANTIS — Compile the file without the MANTIS debugging information. This switch affects F40 programs only.

/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). This switch is not the equivalent of the /P switch of the LOADER, which does not seach any libraries. The /NOSEARCH default is to search the system libraries.

/OLD — Run the appropriate language translator from the system library of old programs (device OLD:), which resides on the disk area [1,3]. If the translator does not exist on device OLD:, try to obtain it from device SYS:.

/REL — Use the existing REL files although newer source files may be present.

/SEARCH — The action is identical to that of the /LIBRARY switch.

/SELF — Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for an individual who keeps a private copy of a translator in his own disk area. System programmers occasionally keep experimental versions of standard translators in their disk area in order to test new features.

/SNOBOL — Compile the file with SNOBOL. Assumed for files with the .SNO extension.

/SYS — Run the appropriate processor from the system library (device SYS:) area of [1,4]. This is the default action.

NOTE

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

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

However, the following is valid.

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

Characteristics

The DEBUG command:

Places your terminal in user mode.
Runs the appropriate processor, LINK-10, and the debugger, destroying your core image.

Examples

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

Generate listings for files FILEA, FILEB, and FILED.

```
.DEBUG TEST↵
MACRO:TEST
LOADING

LOADED 2K CORE
DDT EXECUTION


./        BLT 15,0[16]
```

DELETE Command

Function

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

Command Format

DELETE list

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

If you do not specify a device name or a file structure name, it remains in effect until you change it or until the system reaches the end of the current command string. When you specify a directory name before a file name, the directory becomes the default for all succeeding files. A directory name after a file name applies only to that file.

Characteristics

The DELETE command:

Leaves your terminal in monitor mode.
Runs the PIP program, destroying your core image.

Examples

```
.DELETE TEST.ALG↵
Files deleted:
TEST.ALG

.DELETE PROG1.FOR↵
Files deleted:
PROG1.FOR

.DELETE PROG.REL↵
Files deleted:
PROG.REL
```

DETACH Command

Function

The DETACH command disconnects the terminal from your job without affecting the status of the job. Your terminal is now freed for another job, to either start a new job or attach to a currently running detached job.

An operator can DETACH any device, (except a spooled device.) (See Appendix A of the Operator's Guide.)

Command Formats

1. DETACH

2. DETACH dev: (This format is operator privileged.)

Characteristics

The DETACH command:

Detaches the terminal.

Restrictions

This command is not available to batch users.

Example

```
.DETACH
From Job 30
```

DIRECT Command

Function

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

Command Format

DIRECT output specification = list of input specifications

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

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

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

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

You can truncate switches as long as the resulting abbreviation is unique. Spaces are not permitted with a switch, between the slash at the beginning of the switch and the argument at the end.

Switch	Meaning
/ACCESS:n	Update the access date to the current date for any file of n blocks or less accessed, listed, by the DIRECT program. Because some installations delete files that have not been recently accessed, this switch allows you to prevent such deletion by updating the date. n is interpreted as a decimal number, and refers to the number of blocks actually written in the file unless the /ALLOC switch is also used. If you omit the /ACCESS switch, the system does not change the date. If you specify /ACCESS but you omit :n, n = 5 is assumed.
/ALLOC	List the allocated length of the file instead of the written length. Space on a structure is sometimes allocated in units of more than one block for efficiency. Therefore, the number of blocks allocated to a file may be greater than the number of blocks actually written. The 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. This switch is the complement of the /WRITTEN switch.
/AUTHOR	Output the project-programmer number of the author of the file.
/BEFORE:date:time	List only those files created before the specified date and time. Default is +infinity. Refer to Section 1.4.2.5.
/BLOCKS	Output the length of the file in blocks instead of words. Complement of /WORDS. This is the default.
/CHECKSUM	Compute and print an 18-bit checksum for each file. This checksum is computed by rotating the result left one bit before adding each word. Complement of /NOCHECKSUM.
/DENSITY:n	Use the specified density when reading a magnetic tape. n is 200, 556, 800, 1600, or 6250 bpi. The default is installation-dependent and is modified by the SET DENSITY command.
/DETAIL	Print all available information about a file except for zero values (refer to the extended LOOKUP block in DECsystem-10 Monitor Calls for a list of the values that are printed). The protection and data mode are also listed, even if they are zero. The author is not listed if it is the same as the owner of the directory. Numbers followed by a period are decimal numbers. All other numbers are octal. magnetic tape only.)

DIRECT Command (Cont.)

Switch	Meaning
/EOTS	Stop at the logical end of tape (two consecutive tape marks) when reading a magnetic tape. Complement of /NOEOTS. This is the default.
/FAST	List short form of directory: file name, extension, structure name, and directory name. Equivalent to /F. Complement of /NORMAL and /SLOW.
/FDTA	Get the DTA directory from core.
/FILES:n	Stop after n tape marks (files) when reading a magnetic tape. If you specify /FILES but you omit :n, n = is assumed. Note that the logical EOT will also stop the tape unless /NOEOTS is specified.
/HELP	Type the help text that indicates some of the switches available and how to use them. Equivalent to /H.
/HELP:S	List all switches (S) without their explanations. An asterisk prefixes those switches that have a single-letter abbreviation.
/INDIRECT	Make a CCL file that allows you to sort on directories or files.
/LIST	List the output on device LPT:. Equivalent to /L.
/MARKS	Indicate each tape mark, including the final tape mark, and UFD when reading a magnetic tape. Complement of /NOMARKS.
/NOAUTHOR	Do not output the PPN of the author of the file. This is the default. Complement of /AUTHOR.
/NOCHECKSUM	Do not compute and print the checksum. Complement of /CHECKSUM. This is the default.
/NODETAIL	Do not list the words in the LOOKUP block. Complement of /DETAIL. This is the default.
/NOEOTS	Do not stop at the logical end of tape when reading a magnetic tape. Complement of /EOTS.
/NOFDTA	Do not get the DTA directory from core. This is the default. Complement of /FDTA.
/NOMARKS	Do not indicate each tape mark and UFD when reading a magnetic tape. Complement of /MARKS. This is the default.
/NOPRDEVICE	Do not print the device name. Complement of /PRDEVICE.

Switch	Meaning
/NOPRDIRECTORY	Do not print the directory. Complement of /PRDIRECTORY.
/NORMAL	Output the normal directory listing. This listing includes the file name, extension, length in blocks written, protection, creation date, structure name, non-zero version numbers, and directory name. Complement of /FAST and /SLOW. This is the default. The switch is used to override a /FAST or /SLOW in your option file.
/NOREWINDS	Do not rewind the tape before and after reading a magnetic tape. Complement of /REWINDS.
/NOSORT	Do not produce a file suitable for sorting. Complement of /SORT. This is the default.
/NOSUMMARY	Do not use summary mode and output more than just the summary line. Complement of /SUMMARY. This is the default.
/NOTITLES	Do not output page headers. Complement of /TITLES. This is the default for output to the terminal.
/NOUNITS	Do not list the name of the actual disk unit; instead, just list the structure name. Complement of /UNITS. This is the default.
/OKNONE	Suppress the error message if no files match the wildcard construction.

/OPTION:name	Read 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 Section 1.4.3 for additional information.

/PARITY:ODD /PARITY:EVEN	Specify the parity to be used when reading a magnetic tape. The default is ODD.
/PHYSICAL	Ignore logical names used for device names (refer to the ASSIGN command for a description of logical names).
/PRDEVICE	Print the device.
/PRDIRECTORY	Print the directory.

Switch	Meaning
/PROTECTION:nnn	Give the output file the protection nnn (octal).
/REWINDS	Rewind the magnetic tape before and after reading it. Complement of /NOREWINDS. This is the default.
/RUN:file spec	Run the specified program when this command is finished.
/RUNOFFSET:n	Run the program specified with /RUN with an offset of n. If the switch is omitted, the default is 0; if the switch is given without a value, the default is 1.
/SINCE:date:time	List only those files created after the specified date and time. The default is January 1, 1964. Refer to Section 1.4.2.5.
/SLOW	Output a full listing that includes the file name, extension, length in blocks written, protection, creation date and time, access date, structure name, and directory name. Equivalent to /S. Complement of /FAST and /NORMAL. (Disk and magnetic tape only.)
/SORT	List the file structure name, if there is more than one or if the files are on magnetic tape, and directory name, if a wildcard directory is given, on each line instead of only on the first line in which they change. Multiple spaces are output instead of TABs. Project-programmer numbers include leading zeros; the date is in ANSI format: 19721009 for Oct 9, 1972. This switch is used to prepare a file to be sorted by the SORT program (refer to the DECsystem-10 COBOL Programmer's Reference Manual). Complement of /NOSORT.
/SUMMARY	Output only the summary line that indicates the total number of blocks and files. Note a /FAST/SUMMARY lists a /FAST listing followed by the summary.
/TITLES	Cause a heading to be output on each page consisting of a label for each column, date, time, and page number. Standard output to the line printer has this heading. Complement of /NOTITLES.
/UNITS	List the name of the actual disk unit on which the files are stored instead of the file structure name. Complement of /NOUNITS.
/WIDTH:n	Output several entries on a single line to make the output n columns wide. If /FAST is specified for output to a terminal, four file names appear per line. The default for n is 64 columns.

Switch	Meaning
/WORDS	Output the length of the file in words instead of blocks. Complement of /BLOCKS.
/WRITTEN	Return the written length of the file rather than the allocated length. Complement of /ALLOC. This is the default.

Characteristics

The DIRECT command:

Leaves the terminal in monitor mode.
Runs the DIRECT program, thereby destroying your core image.

Examples

`.DIR DTA003:↵`	Lists all files on DTA3.
`.DIR *.MAC↵`	Lists all files with extension .MAC in all file structures in the user's job search list.
`.DIR TEST.F4[27,4072]↵`	Lists the directory entry for the file TEST.F4 in user area 27,4072.

The ordinary default directory.

```
.DIR↵

GHOST   EXE        16  <055>   28-Jul-77         1(5)    DSKC:    [27,5110
]
COMMAN  RNO         1  <055>    4-Aug-77
LOGOUT  MIC         1  <055>    4-Aug-77
MANUAL  RNO         3  <055>    4-Aug-77
MICV2   DOC        95  <055>    4-Aug-77
PATMON  RNO        27  <055>    4-Aug-77
PRINT   MIC         1  <055>    4-Aug-77
TAPES   RNO       112  <055>    4-Aug-77
UUOSYM  RNO       218  <055>    4-Aug-77
2050    RNO         3  <055>    4-Aug-77
COM2B   RNO       127  <055>    4-Aug-77
COM2C   RNO       105  <055>    4-Aug-77
COM2D   RNO        46  <055>    4-Aug-77
TITLE   RNO         1  <055>    4-Aug-77
COM12   RNO        72  <055>    4-Aug-77
FILDAE  RNO        58  <055>    4-Aug-77
COM1    RNO        98  <055>    4-Aug-77
C                   1  <055>   31-Jul-77
COM2A   RNO       161  <055>    4-Aug-77
COM11   RNO       132  <055>    4-Aug-77
COMM2   RNO       841  <055>    4-Aug-77
COIND2  IND        12  <055>    4-Aug-77
SWITCH  BAK         1  <055>    4-Aug-77
SWITCH  INI         1  <055>    4-Aug-77
HEADS   BAK        16  <055>    4-Aug-77
COM3    MEM       131  <055>    4-Aug-77
COM3    BAK       126  <055>    4-Aug-77
COM3    RNO       126  <055>    4-Aug-77
TOPS10  RNO         4  <055>    4-Aug-77
  Total of 2536 blocks in 29 files on DSKC: [27,5110]
```

DIRECT Command (Cont.)

```
.DIR/ALLOC↵
.
GHOST   EXE       20  <055>   28-Jul-77        1(5)     DSKC:    [27,5110
]
COMMAN  RNO       10  <055>    4-Aug-77
LOGOUT  MIC       10  <055>    4-Aug-77
MANUAL  RNO       10  <055>    4-Aug-77
MICV2   DOC      100  <055>    4-Aug-77
PATMON  RNO       30  <055>    4-Aug-77
PRINT   MIC       10  <055>    4-Aug-77
TAPES   RNO      120  <055>    4-Aug-77
UUOSYM  RNO      220  <055>    4-Aug-77
2050    RNO       10  <055>    4-Aug-77
COM2B   RNO      130  <055>    4-Aug-77
COM2C   RNO      110  <055>    4-Aug-77
COM2D   RNO       50  <055>    4-Aug-77
TITLE   RNO       10  <055>    4-Aug-77
COM12   RNO       80  <055>    4-Aug-77
FILDAE  RNO       60  <055>    4-Aug-77
COM1    RNO      100  <055>    4-Aug-77
C                 10  <055>   31-Jul-77
COM2A   RNO      170  <055>    4-Aug-77
COM11   RNO      140  <055>    4-Aug-77
COMM2   RNO      850  <055>    4-Aug-77
COIND2  IND       20  <055>    4-Aug-77
SWITCH  BAK       10  <055>    4-Aug-77
SWITCH  INI       10  <055>    4-Aug-77
HEADS   BAK       20  <055>    4-Aug-77
COM3    MEM      140  <055>    4-Aug-77
COM3    BAK      130  <055>    4-Aug-77
COM3    RNO      130  <055>    4-Aug-77
TOPS10  RNO       10  <055>    4-Aug-77
  Total of 2720 blocks in 29 files on DSKC: [27,5110]
```

```
.DIR/DETAIL↵

DSKC1:GHOST.EXE[27,5110]
Access date:  4-Aug-77
Creation time, date: 11:33 28-Jul-77
Access protection: 055
Mode: 16
Words written: 2048.
Version: 1(5)
Estimated length: 16.
Blocks allocated: 20.
Written on: Unit(s) 4 on controller 2 on CPU 1026
Data block in directory: 64700.
Internal creation date,time: 28-Jul-77 11:33:38
RIB block number: 23500.
```

DIRECT Command (Cont.)

```
DSKCO:COMMAN.RNO[27,5110]
Access date:  4-Aug-77
Creation time, date: 10:09  4-Aug-77
Access protection: 055
Mode: 1
Words written: 32.
Estimated length: 1.
Blocks allocated: 10.
Written on: Unit(s) 3 on controller 1 on CPU 1026
Data block in directory: 64700.
Internal creation date,time:  4-Aug-77 10:09:07
RIB block number: 59880.

DSKCO:LOGOUT.MIC[27,5110]
Access date:  4-Aug-77
Creation time, date: 10:09  4-Aug-77
.
.
.
```

DISMOUNT Command

Function

The DISMOUNT command allows you to return devices to the monitor's pool of available resources, and to remove a file structure from the search list. Restricted devices are returned to the restricted pool and unrestricted devices to the unrestricted pool. In addition, the DISMOUNT command enables you (via switches) to reassign an unrestricted device to another job, or to retain control of a device you have finished using. When this command is applied to non-file structures it is similar to the DEASSIGN command. DISMOUNT also handles positioning of tapes, reassignment of devices, and FACT file accounting. When this command is applied to file structures, it keeps track of logged-out quotas and, if necessary, enforces them, allows physical removal of disk packs (if there are no other users of pack), and removes the file structure name from the job's search list.

If you are a timesharing user, you may continue the job as soon as the DISMOUNT command is issued even if operator action is required. If you wish to wait for the operator to acknowledge the request before continuing with the job, you must specify the /WAIT switch (see below). A batch job always waits until the DISMOUNT has been completed, unless the /NOWAIT switch has been specified.

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

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

Command Format

```
DISMOUNT dev:switches
```

dev: = any previously ASSIGNed or MOUNTed device or file structure name. The name may be a physical or logical name (unless the /REASSIGN switch is given, below) or it may be NUL:. This argument is required except where otherwise indicated.

/switches = optional switches that can be used to further specify the requested action. Switches are processed in the order specified by the user. If there are any conflicting switches in the command string, the last one scanned is used. (An exception to this is the use of both the /WAIT and /NOWAIT switches within a single command string. If both switches are so specified, a fatal error condition results.) If you specify switches that do not apply to the device type these switches are ignored: (DISMOUNT DSKB:/REWIND). Only as few characters as are necessary to make the switch unique are required.

Switch — Meaning

/CHECK — Check and list this job's pending DISMOUNT requests. This switch is intended to be used without other switches and without a device specification. If other switches or a device appear in the same command string with the /CHECK switch, they will be ignored.

/HELP — Types out a brief description of the DISMOUNT command and the list of switches that are valid with it. This switch is intended to be used without other switches, and without a device specification. If other switches, or a device, appear in the command string with the /HELP switch, they will be ignored.

/KEEP — Retain control of the device for further use. Keep the specified device assigned to this job, even though it is currently dismounted. This switch is intended to be used with non-file structure devices.

/PAUSE — Notifies you (giving you an opportunity to cancel the request) before queuing the DISMOUNT request to the operator.

/REASSIGN:n — Reassign the device to job number n. The device must be a physical device name. Restricted devices may not be reassigned.

/RELEASE — Release conrol of the device, default condition.

/REMOVE — Removes the file structure from your search list. If there are no other jobs using the file structure, and if it does not contain essential files, the operator is notified that you are finished with the device, and it should be removed.

NOTE

> "R" is a valid abbreviation of REMOVE. Therefore, if the /REASSIGN, /RELEASE, or /REWIND are desired, at least the first three characters of those switch names are required.

/REWIND — Rewind the tape. This switch is ignored if the tape has already been rewound or unloaded.

/STAY — Do not reposition the device. This is ignored for disk.

Switch	Meaning
/UNLOAD	Unload the tape, default condition for tapes. NOTE The repositioning switches (/REWIND, /STAY and /UNLOAD) are ignored and the default repositioning occurs when the /KEEP switch is not specified.
/VID:text	A visual identification passed as a comment to the operator. The argument can be up to 50 characters in length and may be delimited by single or double quotation marks. (The quotation marks are included in the 50 characters.)
/WAIT	Wait for the operator to acknowledge the DISMOUNT request before you continue the job. This switch is ignored if operator action is not required to complete the DISMOUNT: a DISMOUNT command for a file structure that others are using. This is the default for batch jobs. Also, the /WAIT switch can be used without a device specification, and without other switches: the job waits for all pending DISMOUNT requests to be completed. /WAIT is the default condition for batch jobs.
/NOWAIT	Does not wait for the operator to acknowledge DISMOUNT requests before allowing you to continue the job. This is the default condition for time-sharing jobs.

Characteristics

The DISMOUNT command:

Places the terminal in user mode.
Runs the UMOUNT program, thereby destroying the user's core image.

Examples

```
.ASSIGN DTA SCRATCH↵
DTA010 ASSIGNED

.DISMOUNT SCRATCH↵
DTA010 DISMOUNTED
```

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

```
.ASSIGN MTA FAILSA↵
MTA010 ASSIGNED

.DISMOUNT FAILSA/KEEP/REWIND↵
MTTA010 DISMOUNTED
```

```
.SYSTAT .↲
 7    10,4072  TTY11    SYSTAT   12+SPY  RN              4$
$ MEANS EXECUTE ONLY

BUSY DEVICES:
DEVICE  JOB     WHY        LOGICAL

MTA010   7       AS        FAILSA
```

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

```
.SYSTAT 7 9↲
 7     10,4072  TTY11   SYSTAT    12+SPY    RN             5$
$ MEANS EXECUTE ONLY

BUSY DEVICES:
DEVICE    JOB     WHY        LOGICAL

MTA010     7      AS         FAILSA

9         10,4072 DET                     3      ^C SW        2

.DISMOUNT FAILSA/REASSIGN:9↲
MTA010 DISMOUNTED

.SYSTAT . 9↲
 7       10,4072   TTY11    SYSTAT    12+SPY       RN        6$
$ MEANS EXECUTE ONLY

 9      10,4072   DET                3            ^C SW        2

BUSY DEVICES:
DEVICE    JOB     WHY      LOGICAL

MTA010     9      AS

.
```

This example shows that job 7 has MTA0 assigned to it, and the logical name FAILSA is associated with MTA0. At this time job 9 has no devices assigned to it. The DISMOUNT command releases MTA0 from job 7 and assigns it to job 9. The association between the logical name FAILSA and MTA0 is no longer in effect for job 7. While job 9 now has MTA0 assigned to it, there is no logical name associated with MTA0 any more. The tape itself is unloaded.

```
.DISMOUNT BLKJ/REMOVE↲
BLKJ DISMOUNTED
```

Here, you knew that the job was the only one using the structure BLKJ, and included a /REMOVE switch, so that the operator would physically 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. Since this command implies /NOWAIT, you will not receive notification of the physical removal of the disk pack.

DISMOUNT Command (Cont.)

```
.DISMOUNT/CHECK↵
1. D JOB7 TTY111 10,4072 1 DISMOUNT BLKJ /R
1 COMMAND IN QUEUE
```

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

```
.DISMOUNT/WAIT↵
NONE PENDING FOR YOUR JOB


.
```

The DISMOUNT/WAIT command causes the job to suspend further processing until all pending dismount requests from it have been completed. Here, at the time that the DISMOUNT/WAIT command was issued, there were no pending dismount requests from this job, so the job returns to monitor command level immediately.

```
.DISMOUNT DSKB/REMOVE↵
% DSKB HAS OTHER USERS - /REMOVE IGNORED
DSKB DISMOUNTED

.R SETSRC↵

*T↵
DSKA:/NOCREATE, DSKC:, DSKN:, FENCE
*^C
```

In this example, you requested that DSKB be removed from the search list and physically removed from the disk drive. DSKB is removed from the job's search list, but since there are other jobs using DSKB, a request to physically remove the pack is not queued to the operator.

DSK Command

Function

The DSK command prints disk usage for the combined structures of the job, since the last DSK command, followed by the total disk usage since the job was initialized (logged in). Disk is typed 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, and m and n are to total number of 128-word blocks and written since the job was initialized.

NOTE

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

Command Format

DSK job

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

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

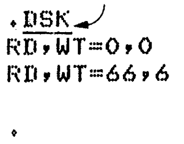
If job is supplied (whether the job of this user or another user) the incremental quantities are not reset to zero.

Characteristics

The DSK command:

Leaves the terminal in monitor mode.

Example



```
.DSK↵
RD,WT=0,0
RD,WT=66,6

.
```

DTCOPY Program

Function

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

1. Copy the entire contents of an input DECtape to an output DECtape.

2. Zero all blocks on an output DECtape and clear the directory.

3. Perform a word-by-word comparison of two DECtapes.

4. Load a bootstrap loader and write it in books 0, 1, and 2 of the output DECtape.

Command Format

```
.R DTCOPY
*output DTA: = input DTA:/switches
```

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

/C Copy all blocks from the input DECtape to the output DECtape.

/H Do not restart the program after a parity error. Output an error message and continue the program.

/H Type the available switches and their meanings.

/L Load the bootstrap loader into a core buffer. DTCOPY expects the loader to be on logical device PTR in the file named BSLDR.REL. Note that DTCOPY must be SAVed if the loader is to be preserved with the DTCOPY core image.

/N Suppress the directory listing.

/T Write the bootstrap loader in blocks 0, 1, and 2 of the output DECtape. This switch accepts, as input from the terminal, a core bank or offset. The loader is offset and then written on the tape.

core bank = nnnK (16K to 256K)
offset = 100 to 777600 octal

/V Verify the similarities of the two DECtapes by performing a word-by-word comparison and typing on the terminal the number of discrepancies discovered.

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

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

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

Characteristics

The R DTCOPY command:

Places the terminal in user mode.
Runs the DTCOPY program and consequently destroys your core image.

Examples

.R DTCOPY↵	Run DTCOPY
*DTA017:=DTA013:↵	Copy the contents of DTA3 to DTA7 and determine if the two DECtapes are the same (default condition). If the DECtapes disagree, the number of discrepancies is typed on the terminal.
*DTA002:/Z↵	Zero all blocks and clear the directory on DTA2.
*^C	Return to monitor mode.
.ASSIGN DSK:PTR:↵	The bootstrap loader must be on logical device PTR.
.RENAME BSLDR.REL=DTBOOT.REL↵	COPY expects the bootstrap loader to be named BSLDR.
.R DTCOPY↵	Run DTCOPY.
*/L↵	Load the bootstrap loader into a core buffer.
*^C	Return to monitor mode.
.SAVE DSK↵ .OSAVE DSK:COPY↵	Save DTCOPY so that the bootstrap loader is preserved with the DTCOPY core image.
.START↵	Start the COPY program.
*DTA5 */T	Write the bootstrap loader in blocks 0, 1, and 2 of DTA5.
TYPE CORE BANK AND OFFSET FOR DTBOOT	Respond with size of core bank or offset.
64K	Size of core bank (64K core bank = 177000 offset, top of core - 1000).
*^C	Return to monitor code.

DUMP Command

Function

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

Command Formats

1. DUMP/command/command/command...

2. DUMP @ dev:file.ext [directory]

3. DUMP

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

Characteristics

The DUMP command:

Leaves the terminal in monitor mode.
Runs the DAEMON and DUMP programs, thereby destroying your core image.

Example

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

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

Function

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

Command Formats

1.
   ```
   R Dump
   /command
   ```

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

NOTE

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

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

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

DUMP Program (Cont.)

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

Command	Argument	Meaning
IRADIX	decimal number	Specifies radix for numbers for input. This command uses decimal to compute the argument. The default is 10 for decimal. The argument must be numeric. If the argument is 0 and is missing, the input radix is set back to its default value.
JUSTIFY	LEFT, CENTER, or CENTER	Specifies the justification of the output in the output field. If the output overflows the output field, the entire output appears; it is not truncated. This switch is used in a one-to-one relationship with the MODE and WIDTH commands. If there are more MODE commands, an argument of LEFT is used. If there are more JUSTIFY commands, they are ignored.
LEFTMARGIN	expression	Sets the left margin of the output file. The default is 0.
LINEPAGE	expression	Specifies the number of lines per output page. This counts all lines including blank lines and titles. The default is 50.
MODES or M	ALL, ASCII, DECIMAL, NULL, NUMERIC, OCTAL, RADIX50, SIXBIT, SOCTAL, or	Selects the type of output. All dumps in all modes. ASCII dumps the word as a single right justified character if bits 0-28 are zero or as 5 left justified ASCII characters if bits 0-28 are nonzero. Non-printing characters print as a space. DECIMAL dumps as a signed decimal number. NULL declares that nothing is to be dumped. NUMERIC dumps as a signed number in the current output radix. OCTAL dumps as half-words separated by a comma (default) and takes 13 positions. RADIX50 dumps in RADIX 50. SIXBIT dumps as one right justified SIXBIT character if bits 0-29 are zero, or

DUMP Program (Cont.)

Command	Argument	Meaning
		6 SIXBIT characters if bits 0-29 are nonzero. SOCTAL dumps as signed octal and suppresses leading zeroes. SYMBOLIC dumps as a symbolic instruction. Any mode specification can appear more than once in the command string. The output is in the same order as the MODE list.
NUMPAGE	expression	Specifies that pages are to be numbered. If expression is 0, page numbering is turned off. If expression is not 0, page numbering begins at page = [expression]. If command is omitted, numbering starts at the first page.
OPTION	:name	Reads your option file (DSK:SWITCH.INI[.] /PHYSICAL) to determine your specified switch defaults for this program. The option name appearing as the value of the switch is the pointer to the line to read in the file. For example, if the file contains the line DUMP:OCT/IRADIX:8/ORADIX:8 then you reference this line by typing to DUMP /OPTION:OCT If SWITCH.INI has a line of the form DUMP/switch/switch then the specified switches override the normal DUMP defaults. For example, if you do not want the address dumped, you can place the line DUMP/ADDRESS:OFF

Command	Argument	Meaning
		in the file. If you want the addresses dumped for a particular dump, you must override the file by giving the command /ADDRESS:ON to DUMP.
ORADIX	decimal number	Specifies radix for numbers for output. The default is 10 for decimal. If number is 0, the standard is used. The argument to this command is decimal and must not be an expression.
OUTPUT or O	file descriptor	Specifies the output file. The defaults are: LPT:, the file name of the input file; the extension .LSD; the user's directory. If a file name is given, the device DSK: is assumed.
RIGHTMARGIN	expression	Sets the right margin of the output file. A field may overflow the right margin if it will not fit between the left and right margins. If ADDRESS is ON, the new line will have an address typed. If a page overflow occurs, a title line may also be printed.
RUN	program	Runs the specified program. This command is equivalent to the R monitor command.
SUPERSEDE		Specifies that the output is to supersede an existing file of the same name, if there is one. The complement of this command is APPEND, which is the default.
SYFILE	file descriptor	Specifies the file to take symbols from if XTRACT command is specified. Defaults are: DSK:, the file name of the input file; one of the saved file extensions; your directory area.

DUMP Program (Cont.)

Command	Argument	Meaning
TDUMP or T	dump descriptor, dump descriptor,...	Dumps specified bytes to both output file and TTY.
TITLE	string of characters	Specifies a title to be included in the subsequent page headings. If no argument is specified, titling is turned off. After this command, an EJECT command should be given to skip to a new page.
TYPE[1]	DAE, DAT, HGH, LOW, SAV, SHR, XPN, EXE	Specifies the format of the input or symbol file. DAE specifies the file is generated by DAEMON. DAT specifies file is a data file (i.e., no special format; therefore, no special processing is done). HGH specifies file in .HGH format; LOW in .LOW format; SAV in .SAV format; SHR in .SHR format; XPN in .XPN format; EXE in .EXE format.
WIDTH	expression	Selects the width of each output mode (see the MODE and JUSTIFY commands). If a MODE command is specified without a corresponding WIDTH, the byte is dumped in exactly the number of positions required followed by 3 blanks. If a WIDTH command is specified, no free blanks are output. If a MODE specification overflows its WIDTH specification, the entire output is given without justification. If expression is omitted, justification is turned off.
XTRACT		Uses the file specified in the last SYFILE command as a core image and extracts the symbol table.

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

An expression is an octal/decimal number, an arithmetic operation. (+, -, *, /, ', or ^ grouped within parentheses), contents operator. ([, \, and @), or a symbol.

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

A dump descriptor is any of the following:

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

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

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

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

   word <pos.size>

   where:

   word = the address of the word desired.

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

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

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

   <FROM byte descriptor> & <TO byte descriptor>

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

Characteristics

The R DUMP command:

Places the terminal in user mode.
Is used with disk monitors only.
Runs the DUMP program, thereby destroying your core image.

DUMP Program (Cont.)

Example

```
.DCORE↵

.R DUMP↵

/XTRACT↵

0 symbols eXTRACTed
/MODE SYM,OCT,DEC↵
/WIDTH 30,15,15↵
/JUSTIFY L,R,R,R,R↵
/OUT TTY:↵
/RIGHTMARGIN 70↵
/IRADIX 8↵
/DUMP [140&144]↵

140/    ADDI   1,2                      271040,000002    24838668290
141/    CAMGE  1,370                    315040,000370    27523023096
142/    JRST   401640                   254000,401640    23085581216
143/    PUSH   17,6                     261740,000006    23882366982
144/    MOVEI  6,13                     201300,000013    17364418571
Z                               000000,000000              0
/EXIT

EXIT

.
```

E (examine) Command

Function

The E command examines a core location in your area (high or low segment).

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

Command Format

E addr

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

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

Characteristics

The E command:

Leaves the terminal in monitor mode.
Requires core.

Example

```
.E 140↵
000140/ 000000 000000    .E↵
000141/ 312000 400033    .E↵
000142/ 000000 000000    .
```

EDIT Command [1]

Function

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

Command Format

EDIT file.ext

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

Characteristics

The EDIT command:

Places the terminal in user mode.
Runs the LINED program, thereby destroying your core image.

Example

```
.EDIT DCORE↵
*
```

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

EOF Command[1]

Function

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

```
MTAn: (MF) =
```

Command Format

```
EOF MTAn:
```

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

Characteristics

The EOF command:

Leaves the terminal in monitor mode.
Runs the PIP program, thereby destroying your core image.

Examples

```
.EOF MTA003:↵

?DEVICE MTA003 NOT AVAILABLE

.EOF MTA013:↵

.
```

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

EXECUTE Command[1]

Function

The EXECUTE command translates the source files you specify if necessary (function of COMPILE command), loads the REL files generated into a core image (function of LOAD command), and begins execution of the program. The system determines the proper language translator 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 via a switch.

This command is equal to a LOAD and START sequence of 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, on in core if the monitor has the TMPCOR feature. 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 Section 1.5).

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

Command Format

EXECUTE list

list 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 1.4.2.4.)

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 1.5.5.)

Switch	Meaning
/ALGOL	Compile the file with ALGOL. Assumed for files with the extension of .ALG.
/BIN	Generate 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 Sections 1.5.2 and 1.5.3.) extension is .REL. This is the default action.
/BLISS	Compile the file with BLIS10. Assumed for files with the extension of .B10 or .BLI.
/COBOL	Compile the file with COBOL. Assumed for files with the extension of .CBL.

[1] This command runs the COMPIL program, which interprets the command before running the appropriate language translator for the LOADER.

Switch   Meaning

/COMPILE

Force a compilation on 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 (e.g., in order to obtain a listing of the compilation) because, normally, the system does not perform compilation if the binary file is newer than the source file.

/CREF

Produce a cross-reference listing file on the disk for each file compiled for later processing by the CREF program. The file name for the listing file follows the standard conventions for determining the name of the output file. (Refer to Sections 1.5.2 and 1.5.3.) The extension 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. You need not issue another command to obtain the listing.

/FOROTS

Load the file with FOROTS, the new FORTRAN object-time system.

/FORSE

Load the file with FORSE, the old FORTRAN object-time system.

/FORTRAN

Compile the file with a FORTRAN compiler. Assumed for files with the extension of .F4 and .FOR and all files with nonrecognizable translator extensions if FORTRAN is the standard translator. You need to specify this switch if the file has a nonrecognizable translator extension and FORTRAN is not the standard translator or is not the current default (e.g., EXE/ALGOL FIL1, -FIL2, FIL3/FORTRAN).

/FUDGE

Create a disk file containing the names of the .REL files produced by the command string. When you issue the FUDGE command, PIP reads this file in order to generate a library .RELfile. (Refer to the FUDGE2 program description.) Arguments to this switch are:

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

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.

Switch	Meaning
	This switch is permanent in the sense that it pertains to all REL files generated by the command string.
/F10	Use the FORTRAN-10 compiler when compiling the associated FORTRAN file. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.
/F40	Use the F40 compiler when compiling the associated FORTRAN file. This should be used as a permanent switch because it is not possible to load F40 and FORTRAN-10 binary files together.
/KA10 /KI10	Designate the machine on which the program will execute once it has been loaded. These switches are needed for FORTRAN-10 programs because the compiler generates different code for the KA10 and KI10 processors. The default is the processor on the computer executing the command.
/LIBRARY	Load the files in library search mode. This mode causes a program file in a special library file to be loaded only if one or more of its declared entry symbols satisfies an undefined global request in the source file. The system libraries are always searched. (Refer to the LOADER and LINK-10 documentation.)
/LINK	Cause the file to be loaded by the LINK-10 linking loader. If you use this switch, you should place it before any file specifications because the COMPIL program may have to generate load-control switches. This is the current default action.
/LIST	Generate a disk listing file for each file compiled. The file name for the listing file follows the standard conventions for determining the name of the output file. (Refer to Sections 1.5.2 and 1.5.3.) The extension is .LST. You can later list these files using the LIST command. If the line printer is being spooled for your job, the system writes the listing files on device LPT and automatically spools them at LOGOUT time. Unless you specify this switch, the system does not generate the listing files.
/LMAP	Produce a loader map during the loading process (same action as /MAP) containing the local symbols.
/LOADER	Cause the file to be loaded by the LOADER.
/MACRO	Assemble the file with MACRO. Assumed for files with extensions of .MAC.

Switch	Meaning
/MACY11[1]	Assemble the file with MACY11. Assumed for files with extensions of .P11.
/MANTIS[2]	Compile the file with the MANTIS debugging information. This switch affects F40 files only.
/MAP	Produce loader maps during the loading process. When this switch is encountered, a loader map is requested from the loader. After the library search of the system libraries, the map is written in your disk area with either the file name specified by you (e.g., /MAP:file) or the default file name MAP.MAP if loading is performed by LOADER or the default name nnnLNK.MAP if it is performed by LINK-10. This switch is an exception to the permanent switch rule in that it causes only one map to be produced even though it appears as a permanent switch.
/NEW	Run the appropriate language translator from the experimental system library (device NEW:) area [1,5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:. (Refer to the following NOTE.)
/NOBIN	Do not generate binary files. Unless you issue this switch, the system does not generate binary files. This switch, when combined with the /CREF or /LIST switch, is useful when compiling programs solely for the purpose of generating listings.
/NOCOMPILE	Complement the /COMPILE switch by not forcing a compilation on a source file whose date is not as recent as the date on the binary file. Note that this switch is not the same as the /REL switch, which turns off all compilation, even if the source file is newer than the REL file. /NOCOMPILE is the default action.
/NOLIST	Do not generate listing files. This is the default action.
/NOMANTIS[2]	Compile the file without the MANTIS debugging information. This switch affects F40 programs only.
/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). This is not equivalent to the /P

[1] MACY11, the PDP-11 assembler for the DECsystem-10, will be recognized as a translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

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

Switch	Meaning
	LOADER switch, that does not search any libraries; the /NOSEARCH switch scans the system libraries.
/OLD	Run the appropriate language translator from the system library of old programs (device OLD:) which resides on the disk area [1,3]. If the translator does not exist on device OLD:, try to obtain it from device SYS:. (Refer to the following NOTE.)
/REL	Use the existing REL files although newer source files may be present.
/SEARCH	The action is identical to that of the /LIBRARY switch.
/SELF	Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for an individual who keeps a private copy of a translator in his own disk area. System programmers occasionally keep experimental versions of standard translators in their disk areas to test new features. (Refer to the NOTE below.)
/SNOBOL [1]	Compile the file with SNOBOL. Assumed for files with an extension of .SNO.
/SYS	Run the appropriate language translator from the system library (device SYS:) area [1,4]. This is the default action.

NOTE

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

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

However, the following is valid:

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

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

EXECUTE Command (Cont.)

Characteristics

The EXECUTE command:

Places your terminal in user mode.

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

Starts the execution of the compiled and loaded program.

Examples

```
.EXECUTE TESTER↵
FORTAN TESTER
LOADING

LINK 2K CORE
EXECUTION
```

FAILSA Program

Function

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

NOTE

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

Command Format

```
.R FAILSA
*/switch
```

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

Switch	Explanation
/A	Advances the magnetic tape one physical file.
/B	Backspaces the magnetic tape one physical file.
/C	Causes FAILSA to continue.

1. On disk-to-tape transfers, if the end of tape is reached before all disk files have been saved, FAILSA requests that you mount a new tape and type /C after the tape has been mounted to direct FAILSA to continue.

2. On tape-to-disk transfers, after FAILSA has typed out the number of the mounted tape and its creation time and date, you type /C if this information is correct and you wish all files on the tape to be restored on disk. This feature aids in preventing accidental restoration of files from the wrong tape.

/D	Transfers control of DDT if it is loaded.
/E	Sets creation date and time word so old files can be eliminated by not being transferred in either direction. The format is as follows:

```
*/Emm/dd/yy,tttt<carriage return>
```

Switch Explanation

where mm is the numerical month, dd is the day of the month, yy is the last two digits of the year, and tttt is the time expressed in 24-hour time. If this switch is not specified, Jan. 1, 1964, 0000 hours is used.

/F Sets access date word. This switch is used to transfer, in either direction, only recently accessed files or to save on tape and then delete from disk not recently accessed files (/K before /S or /U). The format is as follows:

```
*/Fmm/dd/yy/<carriage return>
```

where mm is the numerical month, dd is the day of the month, and yy is the last two digits of the year. If this switch is not specified, Jan. 1, 1964 is used.

/G Enables you to save and restore files from a user's area other than your own. This switch sets the source project, programmer number to the specified number. The format is as follows:

```
*/Gmmm,nnn<carriage return>
```

This changes the single, user project, programmer number switch from that of the FAILSA user to mmm,nnn. The new value is retained until the next /G switch. If you do not issue this switch, your project, programmer number is used.

/H Prints a summary of FAILSA operating procedures on your terminal.

/I Sets the magnetic tape density to the installation standard.

/J Looks for the next trailer record (which marks the end of a save set) and points to the beginning of the next save set on the tape or to the logical end of tape, if there is no save set. In the first case, the tape is positioned in such a way that a tape read command will execute properly or a tape write command will overwrite the existing save set. If there is no save set on the tape, the tape is positioned between the two end of file marks designating the logcal end of tape, so that a succeeding tape write (/S or /U) will execute properly. Use the /A switch to position the tape after MAGRIM and/or SAVE and SSAVE header files.

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

FAILSA Program (Cont.)

Switch — Explanation

/L — Types on your terminal a directory of all your files on the tape. FAILSA checks your project, programmer number and uses it to find the correct area of the tape. Only the file names and extensions are typed. This switch is similar to the /F switch in PIP.

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

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

/O — Enables you to save and restore files to another user's area. This switch sets the destination project, programmer number to the specified number. The format is as follows:

```
*/O xxx,yyy <carriage return>
```

This changes the single-user project-programmer number switch from that of the FAILSA user to xxx,yyy. The new value is retained until the next /O switch. If you do not issue this switch, your project, programmer number is used.

/P — Prints a directory of all files on the tape on logical device LST. The directory is listed by project-programmer number and includes the number of blocks allocated per file, a running total for each area, and creation time/access date information. If the logical device LST is assigned to the disk or a DECtape, the file name FAILSA.DIR is given to the directory file.

/Q — Disables the detailed progress reports given by FAILSA on a single-user restore. Complement of /T. This is assumed unless changed by a /T switch.

/R — Restores all files from the tape to the disk. This switch does not accept an argument because it transfers the entire tape. You must be logged in under [1,2] to use this feature.

/S — Saves disk files on the magnetic tape. Up to 16 arguments can follow this switch, separated by spaces and/or tabs. These arguments are names of file structures or disk devices from which files are to be taken. If you specify no arguments, the system saves the entire disk; in this case, you must be logged in under [1,2].

Switch Explanation

/T Tells you, on a single-user restore, the user area that has been found on the tape, the file that has been selected for transfer to the disk, when the transfer is begun, and when the transfer is completed. An example of this progress report is shown below:

```
LOCATED ON TAKE DSKC 27,235
FOUND ON TAPE DSKC:FILNAME.EXT
< >
```

There may be a possible pause between the time the angle brackets are typed at the beginning and the end of transfer. The /T switch takes effect on the file for which it appears and ends when a /Q switch is given.

/U Saves, on the magnetic tape, the disk files in the UFD of the user last named in a /G switch, or if no /G switch has been given, of whomever is currently logged in. Up to 16 arguments, separated by spaces and/or tabs, can follow this switch. These arguments are names of file structures or disk devices from which the files are to be taken, and, when used, only those structures mentioned that have files for the specified user are saved. If no arguments are given, all your files on all the structures are saved. You are informed if no files are saved from a specified structure.

/V Generates a request to lock your job in core or resets the request. Note that locking may not be accomplished even though the request is given. FAILSA acknowledges if the lock is successful and does not acknowledge if the lock is unsuccessful. This switch is a toggle switch in the sense that its executions alternately set and reset a request. The operator can consult SYSTAT or SYSDPY to determine the state of the switch.

/W Rewinds the magnetic tape with the logical name FAILSA to load point.

/X Extracts the project-programmer numbers from the tape and reproduces output similar to the original TTY output generated by FAILSA during a save. The output contains the names and the project-programmer numbers on the tape. It is placed on logical device LST, and if LST is assigned to the disk or DECtape, the file name FAILSA.DIR is given to the output.

/Y Debugs new features.

/Z Debugs new features.

/2 Sets the magnetic tape density to 200 bits/inch.

/5 Sets the magnetic tape density to 556 bits/inch.

/8 Sets the magnetic tape density to 800 bits/inch.

FAILSA Program (Cont.)

You can be selective as to the files you save or restore. Files are specified either explicitly by giving the file name and extension or implicitly by giving the wildcard construction.

Characteristics

The R FAILSA command:

Places your terminal in user mode.
Runs the FAILSA program, destroying your core image.

Examples

Saving your disk area

```
.MOUNT MTA: /REELID:FAILSAFE↵
REQUEST QUEUED
NO OPER. JOB FOR REQUESTS
WAITING...2 ^C'S TO EXIT
OPERATOR NOTIFIED
WAITING...
FAILSAFE (MTA000:) MOUNTED


^C
^C

.R FAILSAFE↵

REWINDS ARE NOT AUTOMATIC.
FAILSAFE VERSION 10100,47; FOR HELP, READ SYS:FAILSA.HLP, OR TYPE /H
*/W↵
*/U↵
10:47:50                        22 JUNE 77
TAPE # 01 @ 9 CH 800 BPI
NO FILES SAVED FOR 20,1547                DSKA
DSKC    20,1547
NO FILES SAVED FOR 20,1547
                       DSK                DSKB

$SAVE COMPLETED WITH TAPE # 01
10:47:57                        22 JUNE 77
```

FILCOM Program

Function

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

Command Format

```
.R FILCOM
*output dev:file.ext [directory] = input dev(1):file.ext [directory]
     input dev(2):file.ext [directory]
```

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

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

Defaults

1. If the entire output specification is omitted, the output device is assumed to be TTY. However, the equal sign must be given to separate the input and output specifications of the command string.

2. If an output file name is specified, the default output device is DSK.

3. If the output file name is omitted, the second input file name is used, unless it is null. In this case, the file name FILCOM is used.

4. If the output extension is omitted, .SCM is used on a source compare and .BCM is used on a binary compare.

5. If the [directory] is omitted (input or output side), your default directory is assumed.

6. If an input device is omitted, it is assumed to be DSK.

7. If the file name and/or extension of the second input file is omitted, it is taken from the first input file.

8. A dot following the file name of the second input is necessary to explicitly indicate a null extension, if the extension of the first input file is not null. For example, to compare FILE.MAC and FILE. (i.e., with null extension), use the following command string:

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

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

FILCOM Program (Cont.)

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

Switches — Binary Compare

/H — Type list of switches available (help text from device HLP:).

/nL — Specify the lower limit for a partial binary compare (n is an octal number). This switch, when used with the /nU switch, allows a binary file to be compared only within the specified limits.

/Q — When the files are different, print the message ?FILES ARE DIFFERENT, but do not list the differences. This switch is useful when batch control files want to test for differences but do not want the log file filled with these differences.

/nU — Specify the upper limit for a partial binary compare (n is an octal number). This switch, when used with the /nL switch, allows a binary file to be compared only within the specified limits.

/W — Compare files in binary mode without expanding the files first. (Refer to Appendix D.) This switch is used to compare two binary files with ASCII extensions.

/X — Expand SAV files before comparing them in binary mode. This action removes differences resulting from zero compression. (Refer to Appendix D.)

Switches — Source Compare

/A — Compare files in ASCII mode. This switch is used to force a source compare on two ASCII files.

/B — Compare blank lines. Without this switch, blank lines are ignored.

/C — Ignore comments (all text on a line following a semicolon) and spacing (spaces and tabs). This switch does not cause a line consisting entirely of a comment to become a blank line, which is normally ignored.

/H — Type list of switches available (help text from device HLP:).

/nL — Specify the number of lines that determine a match (n is an octal number). A match means that n successive lines in each input file have been found identical. When a match is found, all differences occurring before the match and after the previous match are output. In addition, the first line of the current match is output after the differences to aid in locating the place within each file at which the differences occurred. The default value for n is 3.

Switches Source Compare

/Q Print the message ?FILES ARE DIFFERENT, when the files are different, but do not list the differences.

/S Ignore spaces and tabs.

/U Compare in update mode. This means that the output file consists of the second input file with vertical bars (or backslashes for 64-character printers) next to the lines that differ from the first input file. This feature is useful when updating a document because the changes made to the latest edition are flagged with change bars in the left margin. The latest edition of the document is the second input file.

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

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

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

1. Files with extensions .SHR and .HGH are assumed to be high-segment files. Because the word count starts at 400000, upper and lower limits, if used, must be greater than (or equal to in the case of the lower limit) 400000.

2. Files with extensions .SAV, .LOW, and .SVE are assumed to be compressed core image files and are expanded before comparing.

3. Files with the extension .EXE are assumed to be in .EXE program format.

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

Output

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

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

n)m

where:

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

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

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

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

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

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

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

Characteristics

The R FILCOM command:

Places your terminal in user mode.
Runs the FILCOM program, thereby destroying your core image.

Examples

1. You have the following two ASCII files on disk:

```
.TYPE FILEA.RNO↵


A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
```

```
S
T
U
V
W
Z
Y
Z

.
.TYPE FILEB.RNO


A
B
C
G
H
I
J
1
2
3

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

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

```
.R FILCOM                          Run the FILCOM program.

*=FILEA.RNO,FILEB.RNO              Compare the two files on disk and
                                   output the differences on your
                                   terminal. By default, three
                                   consecutive identical lines
                                   determine a match.
File 1) DSKC:FILEA.RNO[27,4072] created: 0825 15-AUG-1977
File 2) DSKC:FILEB.RNO[27,4072] created: 0826 15-AUG-1977

1)1     D
1)      E
1)      F
1)      G
****
2)1     G
**************
1)1     K
1)      L
1)      M
1)      N
****
```

```
2)1     1
2)      2
2)      3
2)      N
**************
1)1     W
1)      Z
1)      Y
1)      Z
****
2)1     4
2)      5
2)      W
2)      X
2)      Y
2)      Z
**************

%files are different
```

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

```
*/4L = FILEA.RNO,FILEB.RNO↵
File 1) DSKC:FILEA.RNO[27,4072] created: 0825 15-AUG-1977
File 2) DSKC:FILEB.RNO[27,4072] created: 0826 15-AUG-1977

1)1     D
1)      E
1)      F
1)      G
****
2)1     G
**************
1)1     K
1)      L
1)      M
1)      N
****
2)1     1
2)      2
2)      3
2)      N
**************
1)1     W
1)      Z
1)      Y
1)      Z
****
2)1     4
2)      5
2)      W
2)      X
2)      Y
2)      Z
**************

%files are different
```

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

```
.R FILCOM↵

*/U = FILEA.RNO,FILEB.RNO↵
        A
        B
        C
|       G
        H
        I
        J
|       1
|       2
|       3
|
        N
        O
        P
        Q
        R
        S
        T
        U
        V
|       4
|       5
|       W
|       X
        Y
        Z

%files are different
```

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

```
*TTY:=DATESA.REL,GETDAT.REL↵
File 1) DSKC:DATESA.REL[27,4072]        created: 1456 17-MAY-1977
File 2) DSKC:GETDAT.REL[27,4072]        created: 1333 08-JUNE-1977

000002  000213 455765   015212 412526   015001 047243
000005  000213 455765   015212 412526   015001 047243
000011  000000 001507   000000 003172          001463
000013  000001 100300   000002 400015   000003 500315
000014  410000 000000   414343 456064   004343 456064
000015  000001 100307   324156 630000   324157 730307
000016  420000 000000   000003 400055   420003 400055
000017  000002 100217   444164 453246   444166 553051
000020  425164 326045   625755 326165   200631 000120
000021  623263 454300   416264 456263   235007 002163
000022  000001 100316   000002 400041   000003 500357
000023  430000 000000   444171 324445   074171 324445
000024  000002 100343   435744 450000   435746 550343
000025  435756 664441   000001 100161   435757 764520
000026  644500 000000   444171 630000   200471 630000
000027  000001 100325   000001 100300          000025
000030  440000 000000   444554 322100   004554 322100
000031  000003 400011   000001 100026   000002 500037
000032  444164 453263   446555 444171   002431 017312
000035  000000 001530   000000 003213          001463
000036  555164 503256   000002 100226   555166 403070
000037  655500 000000   446555 557132   213055 557132
000040  000001 100370   444164 450000   444165 550370
000041  444171 000000   000003 400045   444172 400045
000042  000001 100161   455644 324441   455645 224520
000043  444171 644142   713244 454357   357335 210215
000044  000002 400041   444500 000000   444502 400041
000045  446356 324570   000002 400021   446354 724551
000046  516400 000000   465162 636432   173562 636432
000047  000002 100433   455664 627100   455666 727533
000050  466257 553264   000002 100352   466255 453136
000051  716045 000000   544541 603246   252504 603246
000052  000001 400015   5441^O

%files are different
```

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

```
.R FILCOM↵


*TTY:=PAT.SHR,XMH.SHR↵
File 1) DSKC:PAT.SHR[27,4072]    created: 0000 23-JAN-1976
File 2) DSKC:XMH.SHR[27,4072]    created: 0836 28-AUG-1976

400000   035064 417316   001754 400010   034730 017306
400002   035063 035777   000137 001777   035154 034000
400003   017532 000000   003161 000000   014453 000000
400004   000000 000000   000201 000000   000201 000000
400005   604164 000000   705550 000000   101434 000000
400007   000000 000000   000400 000000   000400 000000
400010   051140 401256   254040 400011   205100 001247
400011   254000 400027   047000 000000   213000 400027
400012   274640 000002   334740 400013^O

%files are different
```

FILCOM Program (Cont.)

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

```
*^C

.R FILCOM↵


*TTY:=XMH.SHR↵
400000   001754 400010
400001   254200 000000
400002   000137 001777
400003   003161 000000
400004   000201 000000
400005   705550 000000
400006   000000 000000
400007   000400 000000
400010   254040 400011
400011   047000 000000
400012   334740 400013
400013   776422 000140
400014   200640 000017
400015   201600 000000
400016   201700 401533
400017   201700 401533
400020   260740 402314
400021   265040 402756
400022   000041 000041
400023   555757 505460
400024   201040 000001
400025   201100 020321
400026   505100 425453
400027   260740 402323
400030   202055 000002
400031   201040 000001
400032   201100 020321
400033   505100 425453
400034   260740 402323
400035   202055 000001
400036   200455 000001
400037   505440 441200
400040   402015 000005
400041   201040 000001
400042   202055 00^O
```

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

```
.R FILCOM↵

*TTY:/150L/160U=DATESA.REL,GETDAT.REL↵
File 1) DSKC:DATESA.REL[27,4072]        created: 1456 17-MAY-1977
File 2) DSKC:GETDAT.REL[27,4072]        created: 1333 08-JUNE-1977

000150  416200 000000   620000 000000   236200 000000
000151  000001 400045   000001 100262          277563
000152  636544 416445   675762 533221   043226 125664
000155  000000 001634   000000 003317          001463
000156  000002 400051   000002 100325          277524
000157  636544 416445   675762 533241   043226 125604
000160  324570 516400   566300 000000   642670 516400

%files are different
```

FILE Command

Function

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

Command Formats

1. FILE C

   Checks the queue of FILE requests to determine if any of your requests are still pending. No argument is required. Pending requests for your job will be listed.

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

   Deletes the specified files from DECtape. Requires tape-ID and list of file names as arguments. The tape-ID is a 1- to 6-character alphanumeric name that is used to identify the tape. After the files are deleted, an automatic FILE L is performed.

3. FILE F, tape-ID,dev:file.ext[proj,prog],dev:file.ext[proj,prog]...

   Files information onto a DECtape. Requires tape-ID and list of file specifications as arguments. The file specifications may include an explicit file structure name and project, programmer number so that you can copy files from a disk area other than your own. The device and project, programmer number need not be specified for subsequent file specifications if they do not change. That is, only the programmer number (e.g., [,104]) need be specified if the file to be copied has the same project number as yours and only the project number [41,] need be specified if you have files from another project that you wish to copy. 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 it can be updated. 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 copied, an automatic FILE L is performed.

4. FILE L, id

   Reads the directory of a DECtape and places it in your disk area as an ASCII file with the file name tape-ID.DIR. The file is placed on the first writable file structure in your search list provided that that structure contains your UFD. Tape-ID is a 1- to 6-character alphanumeric name that is used to identify the tape. It is the only argument. You may then print the directory with a monitor command. The format of the DECtape directory is similar to the directory file written by the monitor command DIRECT. If errors occur while the FILE command is being processed, error messages are written into the directory file tape-ID.DIR.

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

   Retrieves (transfers) information from your DECtape to the disk. Requires tape-ID and list of file names as arguments.

If the specified files already exist, they are superseded only if their protection 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.

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 typed to you. Control returns when all requests have been processed. You may type two CTRL/Cs (^C^C) if you decide not to wait.

7. FILE Z, TAPE-ID,dev:file.ext[ppn],dev:file.ext[ppn]...

   Zeroes the directory of the DECtape before the files are copied and then performs the same operations as the FILE F command. Requires tape-ID and may have a list of directory specifications arguments. After the files are copied, an automatic FILE L is performed.

The C and W functions are the only requests that are performed whenever possible. Your terminal and job are free to proceed before the request is completed except for batch jobs, which always wait. The function argument is optional. If the function argument is not specified, a brief dialogue is performed.

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 named file exists on more than one structure, the first one in your search list is copied. 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.

The wildcard (*) and wild-character (?) constructions may be used, but care should be taken when generic DSK: is typed (or implied) for FILE R commands. Because DSK: may define many file structures, the single file structure is chosen as follows:

When the wildcard or wild-character construction is used for the file name or extension, the first structure in your search list on which you may create files and has a UFD is used. This is called your standard file structure.

If the wildcard or wild-character construction is not used and the file exists, the first file structure in the search list then contains the specified file is used, unless overridden by a default. (See Examples.) If the file does not exist, the standard structure is used.

WARNING

> If you have a search list with multiple file structures, the asterisk construction (when used with the FILE R command) can cause files to be created, rather than superseded.

When the wild or wild-character 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 wild construction are transferred from all structures in the search list.

A project, programmer number (either explicit or implicit) may only be specified in file descriptors for the FILE F and FILE Z commands.

In the examples below, your search list is as follows:

    SORT:, DSKA: /NOCREATE, DSKB:, DSKB:, DSKC:

You have 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 that area on DSKB and on DSKC.

```
.FILE F,TAPE1,EX.1,EX.2[10,4072]↵
REQUEST QUEUED
1. F JOB30 TTY111 10,4072 TAPE1 DSKB:,DSKA0:EX.1,DSKB0:
     EX.2[10,4072]
1 COMMAND IN QUEUE
.
```

This command requests that the DECtape TAPE1 be mounted, 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 DECtape then be dismounted.

The directory is written on the first structure encountered in the search list which is both writable and on which you have a UFD. In the above 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 only will be copied 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,TAPE2,DSKC:EX.1,[,4072]EX.2↵
REQUEST QUEUED
2. F JOB30 TTY111 10,4072 TAPE2 DSKB:,DSKC0:-
     EX.1,DSKC0:EX.2[10,4072]
2 COMMANDS IN QUEUE
.
```

In this example, you specified the explicit device DSKC for copying from both area. DSKC was typed only once because a device given in a file specification remains in effect for subsequent file specifications unless another device is specified. Note the omission of the project number in the command string. Also note that the p,pn can be specified either before the file name (as in this example) or after the file name (as in the preceding example).

```
.FILE C↵
1, F JOB30 TTY111 10,4072 TAPE1 DSKB:DSKA0:EX.1,
      DSKB0:EX.2[10,4072]
2. F JOB30 TTY111 10,4072 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 R,TAPE1,EX.1↵
REQUEST QUEUED
1. R JOB24 TTY111 10,4072 TAPE1 DSKB:,DSKB:EX.1
1 COMMAND 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 writable file structure in your search list. Note that 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 F,TAPE3,E?.*↵
REQUEST QUEUED
2. F JOB24 TTY111 10,4072 TAPE3 DSKB: SORT:E?.*,
      DSKA:E?.*,DSKB:E?.*,
DSKC:E?.*
2 COMMANDS IN QUEUE
.
```

In cases where a wild-character and/or 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 wild construction, along with each structure in the search list, to OMOUNT.

FILEX Program

Function

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

Command Format

```
.R FILEX
*dev:ofile.ext[directory]<nnn>/switches = dev:
file.ext[directory]/switches
```

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

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

Meaning of Switches:

Help text

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

DECtape Format Specifiers

/F	PDP-15 DECtape format
/M	MIT project MAC PDP-6/10 DECtape format
/O	Old DEC PDP-6 DECtape format
/T	Normal PDP-10 directory format
/V	PDP-11 DECtape format.

File Format Specifiers

/A	ASCII processing; meaningful only for PDP-11 and PDP-15 tapes.
/B	binary processing; overrides default extension. Files read from a PDP-11 format tape with this switch contain four 8-bit bytes in each 36-bit word (1st byte in bits 10-17, 2nd byte in bits

2-9, 3rd byte in bits 28-35, and 4th byte in bits 20-27). Files written on a PDP-11 format tape with this switch are assumed to have the same format.

/C compressed; save file format. This format is assumed for files with extensions .SAV, .LOW, .SVE. The default output extension is .SAV unless the input extension is .LOW or .SVE, in which case the extension remains unchanged.

/D dump format. This format is assumed for files with extension .DMP.

/E expanded core image files (used by FILDDT). This format is assumed for files with extension .XPN. The default output extension is .XPN.

/I image processing; meaningful only for PDP-11 and PDP-15 tapes.

/S simple block (SBLK) format, project MAC's equivalent of .SAV format. The default output extension is .BIN.

DECtape Processing Specifiers

/G (go on), ignores read errors on input device. FILEX checks the always-bad-checksum bit in the 5-series monitor, so this switch is not needed for files with .RPABC on (e.g., CRASH.SAV).

/L (list), causes a directory on an input DECtape file to be typed on your terminal or causes a directory listing of the output DECtape at the end (i.e., after the output).

/P (preserved), causes quick processing (/Q) and preserves the scratch file after processing for use by another command.

/Q (quick), causes an input or output DECtape to be processed quickly by creating a scratch file on the disk. This file is deleted after processing is completed.

/R (reuse), reuses a scratch file preserved by a /P in a previous command.

/Z (zero), causes the appropriate format of a zeroed directory to be written on a DECtape output file. (Zeroing a DECtape directory is equivalent to deleting all the files on the tape.) If TAPEID appears in the output specifier, then TAPEID is written as the tape identifier in the directory. TAPEID is preceded by a circumflex (^) and may be 6 characters on a DECsystem-10 tape, 3 characters on a project MAC tape, and is not present on a PDP-6 tape.

Characteristics

The R FILEX command:

Runs the FILEX program, thereby destroying your core image.

Examples

```
.R FILEX↵
```

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

```
*DSK:=DA\A\TA110:TEST.DMP/C↵
```

```
.R FILEX↵
```

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

```
*DSK:SER015.XPN[10,1]=DSKC:CRASH.SAV[1,4]↵
```

FINISH Command

Function

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

Command Format

FINISH dev

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 the logical name of the controlling terminal is cleared.

Characteristics

The FINISH command:

Leaves your terminal in monitor mode.
Requires core.

Restrictions

You cannot continue your program if the device was INITed, but you can start at the beginning or enter DDT.

Examples

```
.FINISH CDR:↵
.
```

```
.FINISH DTA017:↵
.
```

```
.FINISH LPT:↵
```

FUDGE Command[1]

Function

The FUDGE command creates a library REL file from a temporary file generated by a previous COMPILE, LOAD, EXECUTE, or DEBUG command string containing the /FUDGE switch. The library is created with the REL files in the same order in which they were specified in the command string. (Refer to the MAKLIB program writeup and to LOADER and LINK-10 documentation for descriptions of library REL files.)

NOTE

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

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

Command Format

FUDGE

Characteristics

The FUDGE command:

Leaves your terminal in monitor mode.
Runs the PIP program, thereby destroying your core image.

Examples

```
.COMPILE/FUDGE:LIBRARY/MACRO TEST.MAT,DATAPRO.CBL,SCIENCE.F4↵
```

```
.FUDGE↵
```

Create a disk file named LIBRAR that contains the names of all the REL files produced.

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

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

GET Command

Function

The GET command loads a core image from a retrievable storage device but does not begin execution.

This command clears all of your core. However, programs should not count on this action and should explicitly clear those areas of core that are expected to contain zeros (i.e., programs should be self-initializing). This action allows programs to be restarted by a ^C, start sequence without having to do another GET command.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (2 EOFs in a row).

Command Format

```
GET dev:file.ext[proj,prog] core
```

The arguments and the defaults are the same as in the RUN command.

Characteristics

The GET command:

Leaves your terminal in monitor mode.
Does not operate when the device is currently transmitting data.

Example

```
.GET SYS:PIP↵
JOB SETUP


.GET TEST↵
JOB SETUP
```

GLOB Program

Function

The GLOB program reads multiple binary program files and produces an alphabetical cross-referenced list of all the global symbols (symbols accessible to other programs) encountered. This program also searches files in library search mode, checking for globals, if the program file was loaded by the LOADER in library search mode. (Refer to the LOADER documentation.)

The GLOB program has two phases of operation: the first phase is to scan the input files and build an internal symbol table, and the second, to produce output based on the symbol table. Because of these phases, you can input commands to GLOB in one of two ways. The first way is to specify one command string containing both the output and input specifications. (This is the command string format most system programs accept.) The second is to separate the command string into a series of input commands and output commands.

Command Formats

1. R GLOB

```
outputdev:file.ext[directory] = input dev:file.ext[directory],
     file.ext,...,dev:file.ext[directory] $
```

2. R GLOB

followed by one or more input commands in the form

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

and then one or more output commands in the form

```
outdev:file.ext[directory] = $
```

When you separate your input to GLOB into input commands and output commands (Command Format #2), the input commands contain only input specifications and the output commands, only output specifications. Each output command causes a listing to be generated; any number of listings can be printed from the symbol table generated from the current input files as long as no input commands occur after the first output command. When an input command is encountered after output has been generated, the current symbol table is destroyed and a new one begun.

Defaults

1. If the device is omitted, it is assumed to be DSK. However, if the entire output specification is omitted, the output device is TTY.
2. If the output file name is omitted, it is the name of the last input file on the line (Command Format #1) or is GLOB if the line contains only output commands (Command Format #2). The input file names are required.

3. If the output extension is omitted, .GLB is used. If the input extension is omitted, it is assumed to be .REL unless the null extension is explicitly specified by a dot following the file name.

4. If the project-programmer number [proj,prog] is omitted, your default directory is used.

5. An ALTmode terminates the command input and signals GLOB to output the cross-referenced listing. In other words, a listing is not output until GLOB encounters an ALTmode. The ALTmode appears at the end of the command string shown in Command Format #1 or at the end of each output command shown in Command Format #2.

Switches

Switches control the types of global listings to be output. Each switch can be preceded by a slash, or several switches can be enclosed in parentheses. Only the most recently specified switch (except for L, M, P, Q, and X, which are always in effect) is in effect at any given time. If no switches are specified, all global symbols are output. The following switches are available.

/A Output all global symbols. This is the default if no switches are specified.

/E List only erroneous (multiply defined or undefined) symbols.

/F List nonrelocatable (fixed) symbols only.

/H List the switches available (help text) from HLP:GLOB:HLP.

/L Scan programs only if they contain globals previously defined and not yet satisfied (library search mode).

/M Turn off library search mode scanning resulting from a /L switch.

/N List only symbols which are never referenced.

/P List all routines that define a symbol to have the same value. The routine that defines the symbol first is listed, followed by a plus (+) sign. Subsequent routines that define the symbol are listed, preceded by a plus sign.

/Q Suppress the listing of subsequent definers that result from the /P switch.

/R List only relocatable symbols.

/S List symbols with non-conflicting values that are defined in more than one program.

/X Do not print listing header when output device is not your terminal, and include listing header when it is your terminal. Without this switch, the header is printed on all devices except your terminal. The listing header is in the following format:

```
flags symbol octal value defined in referenced in
```

Symbols listed are in alphabetical order according to their ASCII code values. The octal value is followed by a prime (') if the symbol is relocatable. The value is then relative to the beginning of the program in which the symbol is defined. Flags preceding the symbol are shown below.

M Multiply defined symbol (all values are shown).

N Never referred to (i.e., was not declared external in any of the binary programs).

S Multiply specified symbol (i.e., defined in more than one program but with non-conflicting values). The name of the first program in which the symbol encountered is followed by a plus sign.

U Undefined symbol.

Characteristics

The R GLOB command:

Places your terminal in user mode.
Runs the GLOB program, destroying your core image.

Examples

```
.R GLOB↵
```

Run the GLOB program.

```
*LPT:=MAIN.DTA012:SUB40.SUB500
 <ESC>
```

All global symbols in the program MAIN (on DSK), SUB40, and SUB50 (on DTA2) are listed on the line printer. Along with the symbol is listed its value, the program in which it is defined, all programs in which it is referenced, and any error flags.

```
*DTA014:BATCH.REL,DTA,DTA006:NUMBER.REL,CLASS↵
*DSK:MATH.REL,LIBRARY↵
```

The programs to be scanned are BATCH.REL, DATA.REL on DTA4; NUMBER.REL, CLASS.REL on DTA6; and MATH.REL, LIBRARY.null on DSK.

```
*LPT:=/F4 <ESC>
```

List only nonrelocatable symbols on the line printer.

```
*DSK:SYMBOL=/R<ESC>
```

List only relocatable symbols in the file named SYMBOL in your default directory.

```
*TTY:=/E <ESC>
U EXTSYM SUBTRE
```

Print all erroneous symbols on the terminal. EXTSYM is an undefined symbol appearing in the program SUBRTE.

```
*^C
```

Return to monitor mode.

Function

The GRIPE program accepts text from you and records it in a disk file, thereby enabling other users to record comments and complaints to be read at a later time by the operations staff.

Command Format

R GRIPE

When the GRIPE program responds with a YES?, type the text, using as many lines as necessary, terminated with an ESCAPE. The text is written as a file with <157> protection and includes a header with the date, time, and your project-programmer number. Therefore, you do not need to identify yourself.

Characteristics

The R GRIPE command:

Places your terminal in user mode.
Runs the GRIPE program, thereby destroying your core image.

Example

```
.R GRIPE↵

Yes?  (Depress ESCape key when through)
THIS CONSOLE IS ALMOST OUT OF PAPER AND IT NEEDS A NEW RIBBON.↵
THANK YOU
```

HALT Command

Function

The HALT (^C) command transmits a HALT command to the monitor command interpreter. It stops your job and stores the program counter in the job data area (.JBPC). CTRL/C, unlike HALT, can be issued at user level. (Refer to the DECsystem-10 Monitor Calls for a description of the job data area.)

Command Format

HALT (^C)

Characteristics

The HALT (^C) command:

Places the terminal in monitor mode.
Does not require LOGIN.

Example

```
.^C

.
```

Function

The HELP command is used to obtain useful information on various system features. You can select this information from the following special disk devices: HLP:, NEW:, OLD:, and SYS:.

Command Formats

1. HELP

   Outputs the instructions for receiving information.

2. HELP dev:*

   Outputs both the names of features that have available documentation contained on the specified device and the names of monitor commands, including the SET commands.

3. HELP dev:name

   Outputs the information on the named feature contained on the specified device.

Only the first six characters of the name argument are scanned. These characters must be A through Z, 0 through 9, or asterisk (*).

If the requested file is not found on the specified device, or if the device argument is omitted, the devices are searched in the following order: HLP:, SYS:, NEW:, and OLD:. However, if you have device NEW: enabled in this search list (refer to the SETSRC program writeup) when using Command Format #3, it is searched before device SYS:. Both logical and physical names are searched for each device.

Characteristics

The HELP command:

Leaves your terminal in monitor mode.
Does not require LOGIN.

Examples

```
.HELP *↵

HELP is available for the following:
ALCOMP   ALLOT    ALQUIK   APL      BACKUP   BASIC    BATCH    BLIS10
BLKTRN   BLS36C   BOOT11   BOOTGT   BOX      CDPSPL   CDRSTK   CHANGE
COBDDT   COBOL    COBRG    COPYED   CPL      CREDIR   CREF     CSORT
CSP10    DBMEND   DC72     DC76     DDT      DELFIL   DIRECT   DISMOU
DSKRAT   DTADIR   DTELDR   ERSATZ   FAILSA   FGEN     FILCOM   FILDAE
FILE     FILEX    FIND     FNDBLD   FORDDT   FORDML   FORTRA   FRS
FUDGE2   GALAXY   GALNEW   GASP     GLOB     GRIPE    HELP     HERMES
HYPHEN   INI      INITIA   ISAM     JQ       KILL     LDVT20   LIBARY
LINED    LINK     LOGIN    LOGOUT   LPTSPD   LPTSPL   MACRO    MACY11
MAIL     MAKLIB   MAKVFU   MATCH    MATHLA   MCSGEN   MIMIC    MODUNV
MOUNT    MTCOPY   NETLDR   NETOPR   OLD      OMOUNT   OPSER    PIP
PLEASE   PLTSPL   PMAKER   PPNLST   PTPSPL   QDMP     QDUMP    QUEUE
QUOLST   REDUCE   RERUN    REV      RSXFMT   RUNINP   RUNOFF   SAFEIO
SCDCP    SENGEN   SETSRC   SIMCOM   SIMDDT   SIMDIR   SIMRTS   SIMULA
SORT     SORTC    SOS      SOUP     SPACE    SPELL    SPRINT   SPROUT
SPY      SUBNEW   SUBSYS   SY       SYSDPY   SYSERR   SYSTAT   TAPE
TECO     TECOH    TRACK    TTT3D    TTYTST   UFDREN   UMOUNT   WHEEL
XTEN     1984
```

```
The monitor has the following commands:
ACCOUN  ALLOCA  ASSIGN  ATTACH  BACKSP  BACKTO  CCONTI  CLOSE
COJOB   COMPIL  CONTIN  COPY    CORE    CPUNCH  CREATE  CREF
CSHIFT  CSTART  CTEST   D       DAYTIM  DCORE   DDT     DEASSI
DEBUG   DELETE  DETACH  DIRECT  DISMOU  DO      DSK     DUMP
E       EDIT    EOF     ERROR   EXECUT  FILE    FINISH  FREE
FUDGE   GET     GOTO    HALT    HELP    I       IF      INITIA
JCONTI  KJOB    LABEL   LET     LIST    LOAD    LOCATE  LOGIN
MAKE    MIC     MOUNT   NODE    NOERRO  NOOPER  NSAVE   NSSAVE
ON      OPERAT  OSAVE   OSSAVE  PJOB    PLEASE  PLOT    PRESER
PRINT   PROTEC  PUNCH   QUEUE   R       REASSI  REENTE  RENAME
RESOUR  REVIVE  REWIND  RUN     SAVE    SCHEDU  SEND    SESSIO
SET     SILENC  SKIP    SOS     SSAVE   START   SUBMIT  SYSTAT
TECO    TIME    TPUNCH  TTY     TYPE    UNLOAD  USESTA  VERSIO
WHENEV  WHERE   XCHNGE  ZERO

The monitor has the following SET commands:
BLOCKS  BREAK   CDR     CORMAX  CORMIN  CPU     CTEST   DATE
DAYTIM  DEFAUL  DEFER   DENSIT  DSKFUL  DSKPRI  HOSTES  HPQ
MEMORY  NODEFE  NOMESS  OPR     PHYSIC  RETRY   SCHEDU  SPOOL
TIME    TTY     VIRTUA  VMMAX   WATCH

The monitor has the following SET TTY commands:
ALTMOD  BLANKS  COPY    CRLF    DEBREA  ECHO    ELEMEN  FILL
FORM    GAG     HOLD    IGNORE  LC      NO      PAGE    RTCOMP
SLAVE   SPEED   TABS    TAPE    TIDY    UC      WIDTH

The monitor has the following SET WATCH commands:
DAY     MTA     READS   RUN     VERSIO  WAIT    WRITES


.HELP NEW:*

The monitor has the following commands:
ACCOUN  ALLOCA  ASSIGN  ATTACH  BACKSP  BACKTO  CCONTI  CLOSE
COJOB   COMPIL  CONTIN  COPY    CORE    CPUNCH  CREATE  CREF
CSHIFT  CSTART  CTEST   D       DAYTIM  DCORE   DDT     DEASSI
DEBUG   DELETE  DETACH  DIRECT  DISMOU  DO      DSK     DUMP
E       EDIT    EOF     ERROR   EXECUT  FILE    FINISH  FREE
FUDGE   GET     GOTO    HALT    HELP    I       IF      INITIA
JCONTI  KJOB    LABEL   LET     LIST    LOAD    LOCATE  LOGIN
MAKE    MIC     MOUNT   NODE    NOERRO  NOOPER  NSAVE   NSSAVE
ON      OPERAT  OSAVE   OSSAVE  PJOB    PLEASE  PLOT    PRESER
PRINT   PROTEC  PUNCH   QUEUE   R       REASSI  REENTE  RENAME
RESOUR  REVIVE  REWIND  RUN     SAVE    SCHEDU  SEND    SESSIO
SET     SILENC  SKIP    SOS     SSAVE   START   SUBMIT  SYSTAT
TECO    TIME    TPUNCH  TTY     TYPE    UNLOAD  USESTA  VERSIO
WHENEV  WHERE   XCHNGE  ZERO

The monitor has the following SET commands:
BLOCKS  BREAK   CDR     CORMAX  CORMIN  CPU     CTEST   DATE
DAYTIM  DEFAUL  DEFER   DENSIT  DSKFUL  DSKPRI  HOSTES  HPQ
MEMORY  NODEFE  NOMESS  OPR     PHYSIC  RETRY   SCHEDU  SPOOL
TIME    TTY     VIRTUA  VMMAX   WATCH

The monitor has the following SET TTY commands:
ALTMOD  BLANKS  COPY    CRLF    DEBREA  ECHO    ELEMEN  FILL
FORM    GAG     HOLD    IGNORE  LC      NO      PAGE    RTCOMP
SLAVE   SPEED   TABS    TAPE    TIDY    UC      WIDTH

The monitor has the following SET WATCH commands:
DAY     MTA     READS   RUN     VERSIO  WAIT    WRITES
```

```
.HELP DIRECT↵

Type out=input,input,...
/ACCESS:n = access all listed files under n blocks long
/ALLOCATED = give allocated lengths
/AUTHOR = output author of file
/BEFORE:DATE:TIME = just files created before this date-time
/BLOCKS = give lengths in blocks (default)
/CHECKSUM = compute checksum of each file
/DETAIL = everything from extended LOOKUP
/F = fast mode
/FDTA = get DTA directory from core
/H = this text
/L = out to LPT
/N = normal mode
/NOAUTHOR = don't output author (default)
/NOCHECKSUM = omit checksums (default)
/NODETAIL = don't use detail mode (default)
/NOFDTA = don't get DTA directory from core (default)
/NOPRDEVICE = never print device
/NOPRDIRECTORY = never print directory
/NOSORT = omit sort options (default)
/NOSUMMARY = don't use /SUMMARY mode (default)
/NOTITLES = omit titles (default if TTY:)
/NOUNITS = don't use /UNITS mode (default)
/PRDEVICE = always print device
/PRDIRECTORY = always print directory
/S = slow mode
/SINCE:DATE:TIME = just files created since this date-time
/SORT = output in format amenable to sorting
/SUMMARY = just print summary line
/TITLES = include titles (default if not TTY:)
/UNITS =give actual units for structures
/W:n = try to fill paper width of n columns
/WORDS = give lengths in words
/WRITTEN = give lengths written (default)

  * is wild name, etc.
  ? is wild letter of name, etc.
"out=" may be omitted
default is TTY:.DIR=DSK:*.*[my directory]

on magnetic tape,
/DENSITY:200 or 556 or 800 = select the density
/EOTS = stop at double file mark (DEFAULT)
/FILES:n = stop after doing n files (tape marks)
/MARKS = note each end of file mark
/NOEOTS = don't stop at double file mark
/NOMARKS = don't annotate file marks (default)
/NOREWINDS = don't rewind tape
/PARITY:EVEN = read tape in even mode
/PARITY:ODD = read tape in odd mode (default)
/REWINDS = rewind tape before and after (default)
```

HELP Command (Cont.)

```
.HELP INITIA↵

The INITIA command can be followed by one or more keywords:

BATCH           indicate which, if any, batch system is running
HELP            type this text
KSYS            indicate when time-sharing will cease
NAME            (default) type system name
NORUN           suppress running any operator program
NOTICE          type general operator notice
SETTTY          (default unless logged in) setup terminal
STRUCTURES      type any operator notes about mounted structures
TERMINET        set tabs for a terminet-300
TEXT            TYPE QUICK OPERATOR NOTICE
TTY             type tty mode settings for this terminal

NO may preceed any of the above to override default settings


.
```

Function

The INITIA command performs standard system initialization for the terminal issuing the command. This command is issued automatically at system startup and at the 400 series restart at certain designated terminals, you may reissue it at any time. This command is used to initiate specific system programs, such as the operator service program, OPSER, on a particular console.

When INITIA runs, it finds the file with the name TTY.INI, which is an ASCII file created by you. TTY.INI contains installation-specified options indicating the characteristics of the terminal(s) to be initialized. If TTY.INI cannot be found, INITIA cannot initialize any devices and a start-up message is printed.

Command Format

INITIA or INITIA arg

The command word INITIA can be followed by one or more keywords:

BATCH	Types out the name of the batch that is running on the system (e.g., GALAXY).
HELP	Types a list of available keywords and their meanings.
KSYS	Indicates when timesharing will cease.
NAME	Types the system name (default).
NONAME	Suppresses the typing of the system name.
NORUN	Suppresses the execution of any operator program.
NOTICE	Types the general operator notice.
SET TTY	Set up your terminal (default condition unless you have already logged in).
STRUCTURES	Types any operator notes about mounted structures.
TERMINET	Sets tab spacing identical to a TERMINET-300.
TEXT	Type quick operator notice.
TTY	Types the TTY mode settings for this terminal.

Characteristics

The INITIA command:

Leaves your terminal in monitor mode.
Runs a specific system program.
Does not require LOGIN.

INITIA Command (Cont.)

Examples

```
     .INITIA↵

         RK3A1A KL10 SYS#1026 14:52:34 TTY373 system 1026
     Connected to Node NOVA(31) Line # 41
     Job 35   User COSTLEY,B   [27,5110]
     DSKB:    Green DSKB
     DSKC:    Green DSKC
     DSKN:    System 1026
     YELB:    Yellow DSKB
     YELC:    Yellow DSKC

     .INITIA BATCH↵

         RK3A1A KL10 SYS#1026 14:52:51 TTY373 system 1026
     Connected to Node NOVA(31) Line # 41
     Job 35   User COSTLEY,B   [27,5110]
              GALAXY SPOOLING AND BATCH SYSTEM
     DSKB:    Green DSKB
     DSKC:    Green DSKC
     DSKN:    System 1026
     YELB:    Yellow DSKB
     YELC:    Yellow DSKC

     .INITIA STRUCTURES↵

         RK3A1A KL10 SYS#1026 14:53:12 TTY373 system 1026
     Connected to Node NOVA(31) Line # 41
     Job 35   User COSTLEY,B   [27,5110]
     DSKB:    Green DSKB
     DSKC:    Green DSKC
     DSKN:    System 1026
     YELB:    Yellow DSKB
     YELC:    Yellow DSKC

     .^C
```

JCONTINUE Command

Function

The JCONTINUE command forces a continue of the specified job if the job was in a ^C state because of a call to the device error message routine (HNGSTP).

Command Format

```
JCONTINUE n

     n is the number of the job to be continued.   This  argument
     is required.
```

Characteristics

The JCONTINUE command:

```
     Places your terminal in monitor mode.
     Does not require LOGIN.
```

Example

```
.JCONT 14↵
```

KJOB Command

Function

The KJOB command:

Stops all assigned I/O devices and returns them to the monitor pool.
Returns all allocated core to the monitor pool.
Returns the job number to the pool.
Leaves the console in monitor mode.

Command Formats

KJOB/switch

When you issue the KJOB command, the system checks to see if GALAXY is running. If it is not, refer to the second command format description below. If it is, the system runs the LOGOUT program, which accepts four switches. These switches are listed and described below. The switches are /FAST, /NOMESSAGE, /BATCH, and /HELP.

FAST — to log out immediately and save all files (including temporary files) as they are. This command is equal to an R LOGOUT command or a RUN monitor call to LOGOUT. Unpreserved temporary files (e.g., nnn???.TMP) are deleted where nnn is your job number and the files reside in your disk area. This is the default action of the KJOB command.

BATCH — to delete no files except when you are over the logged-out quota, then delete enough files to be below quota. The files are deleted in the following order: 1) unprotected files according to the category of the file; 2) spooled files not previously queued; and 3) protected files according to the category of the file. The categories of files are as follows: 1) temporary files, 2) relocatable files, 3) backup files, 4) save files, and 5) all other files.

HELP — to list the KJOB options and their meanings.

NOMESSAGE — this switch causes the same action as the /FAST switch, except that no output is printed at the terminal; i.e., all output is suppressed. This action is the default action for a detached job.

The second format (described here) for the KJOB command is used when your system does not include GALAXY.

The command responds with

```
CONFIRM:
```

you may type CTRL/C to abort the logout, or you may type one of the following letters (optionally followed by a space and a list of file structure names):

F to logout immediately and save all files (including temporary files) as they are. This command is equal to an R LOGOUT command or a RUN monitor call to LOGOUT.

D to delete all files on the specified file structures. It responds with DELETE ALL FILES? Type Y or D for YES, any other character for NO.

K to delete all unprotected files (i.e., files with Oxx protection codes) on the specified file structures. If project 1 or other jobs are logged in with the same project-programmer number, the K command responds with DELETE UNPROTECTED FILES? Type Y or K for YES, any other character for NO.

P to save and protect (i.e., assign a protection code of 1 in the owner's field) all but temporary files (TMP, CRF, LST) on the specified file structures. If project 1 or other jobs are logged in with the same project-programmer number, the P command responds with DELETE TEMPORARY FILES? Type Y or P for YES, any other character for NO.

S to save without protecting all but temporary files on the specified file structures. If project 1 or other jobs are logged in with the same project-programmer number, the S command responds with DELETE TEMPORARY FILES? Type Y or S for YES, any other character for NO.

L to list the directories of the specified file structures.

I to individually determine what to do with all files on the specified file structure. After each file name is listed type:

P to protect the file.
S to save the file.
K to delete the file.
Q to learn if over logged-out quota on this structure. If not over quota, nothing is typed, and the same file name is repeated.
E to skip to the next file structure and save this file if below logged-out quota for this file structure. If not below logged-out quota, the message ?file structure name LOGGED OUT QUOTA n EXCEEDED BY m BLOCKS is typed and the same file name is repeated.
H to list responses and meanings.

U to individually determine what to do with all but protected files. Protected files are always preserved. After each file name is listed, type one of the letters associated with the I command above.

B to delete no files except when user is over the logged out quota, then delete enough files to be below quota.

Q to learn if over logged out quota on the specified file structure.

H to list the KJOB options and their meanings.

W to list the names of the files that are deleted.

X to turn off the listing of names of the files that are deleted. Complement of W.

If no file structure names are specified, the responses are for all file structure names in the job search list. If file structure names are specified, the responses apply to those file structures, and CONFIRM is retyped. The KJOB command ignoes all logical assignments.

You have the option of going through the CONFIRM dialogue, even if other jobs are logged-in under the same project-programmer number or if you are logged in under project 1. (However, if sufficient responses are included on the KJOB command line or in a temporary file entered through an alternate entry point, CONFIRM is not typed). By responding to a CONFIRM message, you have an opportunity to organize this disk area by deleting or preserving specific files.

The KJOB program calls the QUEUE program to perform the queueing of files which have been deferred to logout time. This includes all spooled output not previously queued. Queuing may be suppressed with the /Z response (see below).

KJOB

CONFIRM:

When the CONFIRM: response is typed, you may type any one of the above-described letters followed by a space, and an optional file structure name or a list of file structure names separated by commas.

KJOB log file descriptor = /letter list

the log file descriptor has the following format:

dev:file.ext[proj,prog]

If the log file is not a disk or spooled device, TTY is used, and the message %LOG FILE DEVICE IS NOT DSK OR SPOOLED; WILL USE TTY is typed.

letter is any letter from the above-described list. In addition, the following options are available to jobs using this command format:

/Z:n specifies the degree of queueing desired:

n = 0 suppresses all normal queuing done at LOGOUT time.
n = 1 queues only the log file.
n = 2 queues the log file and spooled output.
n = 3 queues the log file, spooled output, and *.LST.
n = 4 queues the log file, spooled output, *.LST, and requests deferred to LOGOUT time (deferred requests are not yet implemented).

If Z is given without a value or if there are no spool bits set for this job, Z:0 is assumed. Otherwise, /Z:2 is assumed if /Z is not typed.

/VL:n specifies that the limit of pages for LPT files is to be n (decimal).

/VC:n specifies that the limit of cards for CDP files is to be n (decimal).

/VT:n specifies that the limit of feet of paper tape for PTP files is to be n (decimal).

/VP:n specifies that the limit of minutes for PLT files is to be n (decimal).

/VR:n specifies that the priority of the queue request is to be n; n is from 0 through 62 (decimal). /VR:10 is the standard or default.

VS:n specifies that the sequence number for the queue request is to be n (decimal).

VD:n specifies that the file disposition of the log file is to be v.

v = D deletes the log file after printing.
v = P preserves the log file after printing.
v = R renames the log file before printing to the queue area and deletes it after printing.

The default is /VD:D.

The letters must appear on the input side of the command string. If the log file is specified, all TTY output is appended to the log file. If no log file is specified or if the log file is not a disk or a spooled device, the default is TTY. In addition, if responses to CONFIRM are required and are not specified on the KJOB command line, these responses will then be read from TTY. Therefore, users should be careful when employing this command format.

Any CONFIRM option that would ask fro re-confirmation (D always, and sometimes K, S, and P) must be repeated if it is on the KJOB command line).

The KJOB program may be entered at the CCL entry point through the RUN monitor call. When this is done, TMPCOR file KJO or disk file nnnKJO.TMP where nnn is your job number in decimal, is used instead of the TTY input.

Characteristics

The KJOB command:

Deassigns your terminal.
Stops all assigned I/O devices because it does not operate when a device is currently transmitting data.
Runs the KJOB and LOGOUT programs.

KJOB Command (Cont.)

Examples

```
 ^C

.KJOB↵
Job 21  User COSTLEY,B [27,5110]
Logged-off TTY41  at 16:57:08  on 29-Jul-77
Runtime: 0:00:04, KCS:62, Connect time: 4:03:26
Disk Reads:701, Writes:29, Blocks saved:25

.K/F↵
Job 34  User CUSTER,L. [27,4072]
Logged-off TTY34  at  8:43:55  on  9-Aug-77
Runtime: 0:00:01, KCS:19, Connect time: 0:13:39
Disk Reads:62, Writes:11, Blocks saved:4840
```

LABEL Command[1]

Function

The LABEL command allows you to write an identifier onto a DECtape. This command is implemented for your convenience. The identifier is stored on the tape itself and is displayed when you list a directory of the tape with the DIRECT command. The PIP and FILEX programs can also read and write DECtape labels. As a user you should assign a unique label to each DECtape in order to avoid confusing one tape with another.

Command Format

LABEL dev: ^name^

dev: is a physical or logical name which represents a DECtape.

^ is the delimiter of the DECtape identifier, which may be any non-alphanumeric character.

name is a 1- to 6-character SIXBIT name to be used as the identifier. Any characters can be used except the delimiter.

If the identifier consists entirely of alphanumeric characters, the delimiters can be omitted.

The DECtape must be assigned to you before this command is executed.

Characteristics

The LABEL command:

Leaves your terminal in monitor mode.
Runs the PIP program, thereby destroying your core image.

Examples

```
.LABEL DTA007:COBOL:1↵

.LABEL DTA003:^PIP.33^↵

.ASSIGN DTA002:LOGNAM↵

.LABEL LOGNAM:^TAPID^↵

.
```

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

LIST Command[1]

Function

The LIST command directs PIP to list the contents of named source file(s) on the line printer (LPT). The output goes either to LPT immediately or to the disk to be spooled to LPT if it is being spooled for this job. (Refer to the QUEUE and PRINT commands.) If the LPT is being spooled, the QUEUE program should be used instead of the LIST command since it saves time and disk accesses.

Command Format

LIST list

list 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 on to PIP.

Characteristics

The LIST command:

Leaves your terminal in monitor mode.
Runs the PIP program, thereby destroying your core image.

Examples

```
.LIST TEST.*↵

.LIST *.MAC↵


.LIST DTA004:A,B,C↵

.
```

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

LOAD Command[1]

Function

The LOAD command translates the specified source files if necessary (function of COMPILE command), runs the linking loader, and loads the .REL files generated. The language translator used is determined by the source file extension or by switches in the command string. (Refer to the COMPILE command.) If a REL file already exists with a more recent date than that of the source file, compilation is not performed (unless requested via a switch).

This command generates a core image but does not begin execution. At this point, 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 command with its arguments is remembered in a temporary file on disk, or in core if the monitor has the TMPCOR feature. Therefore, issuing one of these commands without any arguments causes the arguments saved in the temporary file to be reused. (Refer to Section 1.5.)

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

Command Format

LOAD list

list 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 1.4.2.4.)

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

Switch	Meaning
/ALGOL	Compile the file with ALGOL. Assumed for files with the extension of .ALG.
/BIN	Generate a binary file for each file compiled. The name for the binary file follows the standard conventions for determining the name of the output file. (Refer to Sections 1.5.2 and 1.5.3.) The extension is .REL. This is the default action.

[1] This command runs the COMPIL program, which interprets the command before running the appropriate language translator and linking loader.

LOAD Command (Cont.)

Switch Meaning

/BLISS[1] Compile the file with BLIS10. Assumed for files with the extension of .B10 and .BLI.

/COBOL Compile the file with COBOL. Assumed for files with the extension of .CBL.

/COMPILE Force a compilation of this file even if a binary file exists with a newer date and time than the source file. This switch is used to obtain an extra compilation (e.g., in order to obtain a listing of the compilation) because, normally, compilation is not performed if the binary file is newer than the source file.

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

/FOROTS Load the file with FOROTS (the new FORTRAN object-time system).

/FORSE Load the file with FORSE (the old FORTRAN object-time system).

/FORTRAN Compile the file with a FORTRAN compiler. Assumed for files with the extension of .F4 and .FOR and all files with nonrecognizable translator extensions if FORTRAN is the standard translator. This switch is needed if the file has a nonrecognizable translator extension and FORTRAN is not the standard translator or is not the current default (e.g., LOAD/COBOL, TEXT1, TEXT2, TEXT3/FOR).

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

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

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

[1] BLIS10 will be recognized as a language translator only if the appropriate assembly switch is set. However, this assembly switch setting is not supported.

Switch Meaning

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

[proj,prog] - the directory in which to place the file. Your directory is assumed if none is given.

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

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

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

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

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

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

/LIST Generate a disk listing file for each file compiled. The file name for the listing file follows the standard conventions for determining the name of the output file. (Refer to Sections 1.5.2 and 1.5.3.) The extension is .LST. These files can be listed later with the LST command. If the line printer is being spooled for this job, the listing files are written on device LPT and are automatically spooled at LOGOUT time. Unless this switch is specified, listing files are not generated.

LOAD Command (Cont.)

Switch — Meaning

/LMAP — Produce a loader map during the loading process (same action as /MAP) containing the local symbols.

/LOADER — Cause the file to be loaded by the LOADER.

/MACRO — Assemble the file with MACRO. Assumed for files with extension of .MAC.

/MACY11 — Assemble the file with MACY11. Assumed for files with an extension of .P11.

/MANTIS — Compile the file with MANTIS. This switch affects F40 FORTRAN programs only.

/MAP — Produce a loader map during the loading process. When this switch is encountered, a loader map is requested from the loader. After the library search of the default libraries, the map is written in your disk area with either the file name specified by you (e.g., /MAP:file) or with the default file name MAP.MAP if loading is performed by LOADER or the file name nnnLNK.MAP if loading is performed by LINK-10. This switch is an exception to the permanent compile switch rule in that it causes only one map to be produced although it may appear as a permanent switch.

/NEW — Run the appropriate language translator from the experimental system library (device NEW:) area [1,5]. If the translator does not exist on device NEW:, try to obtain it from device SYS:. (Refer to the following NOTE.)

/NOBIN — Do not generate binary files. Unless this switch is given, binary files are generated. This switch, when combined with the /CREF or /LIST switch, is useful when compiling programs solely for the purpose of generating listings.

/NOCOMPILE — Complement the /COMPILE switch by not forcing a compilation on a source file whose date is not as recent as the date on the binary file. Note that this switch is not the same as the /REL switch which turns off all compilation, even if the source file is newer than the REL file. /NOCOMPILE is the default action.

/NOLIST — Do not generate listing files. This is the default action.

Switch Meaning

/NOMANTIS Compile the file without the MANTIS debugging information. This switch affects F40 programs only.

/NOSEARCH Load all routines of the file whether the routines are referenced or not. Since this is the default action, this switch is used only to turn off libary search mode (/LIBRARY). This is not equivalent to the /P LOADER switch because /P does not search any libraries where /NOSEARCH will scan the default libraries.

/OLD Run the appropriate language translator from the system library of old programs (device OLD:) that resides on the disk area [1,3]. If the translator does not exist on device OLD:, try to obtain it from device SYS:. (Refer to the following NOTE, after /SYS.)

/REL Use the existing .REL files although a newer source file may be present.

/SEARCH The action is identical to that of the /LIBRARY switch.

/SELF Run the appropriate language translator from device DSK: instead of from the system library (device SYS:). This switch is useful for anyone who keeps a private copy of a translator in his own disk area. System programmers occasionally keep experimental versions of standard translators in their disk area in order to test new features. (Refer to the following NOTE, after /SYS.)

/SNOBOL[1] Compile the file with SNOBOL. Assumed for files with an extension of .SNO.

/SYS Run the appropriate language translator from the system library (device SYS:) area [1,4]. This is the default action.

NOTE

> Once a language translator has been specified from a particular area (e.g., /SELF), it cannot be called from a different area within the same command string. The following is illegal:
>
> ```
>      .LOAD ITEM.CBL/SYS.ITEM02.CBL/SELF
> ```
>
> However, the following is valid:
>
> ```
> .COMPIL ITEM.CBL/SYS
> .COMPIL ITEM02.CBL/SELF
> .LOAD /REL ITEM, ITEM02
> ```

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

LOAD Command (Cont.)

Characteristics

The LOAD command:

Leaves your terminal in monitor mode.
Runs the appropriate language translator and linking loader, thereby destroying your original core image.

Examples

```
.LOAD TEST↵
MACRO:  TEST
LOADING

LOADER 2K CORE

EXIT

.
```

LOCATE Command

Function

The LOCATE command logically establishes your job at a specified network node. When your job is initiated, your logical station corresponds to your physical location. Therefore, this command is needed only if you wish to change your logical node/station. The main use of this command is to change the default device list of the job. For example, if you wish to have the devices at remote station CHRIS perform all I/O, you can issue the following command:

```
.LOCATE CHRIS
```

and the default I/O devices will be those on node CHRIS.

Command Format

LOCATE node-id

node-id is the logical identifier of the node upon which you wish to be located.

An argument of 0 locates the job at the node where the job's command decoder is. A null argument implies the node/station of your terminal, i.e., his physical node/station.

Characteristics

The LOCATE command:

Leaves your terminal in monitor mode.
Is valid only in networks or in systems with a remote station.

Restrictions

The LOCATE command must specify a node/station that is currently in contact with the command decoder node/station.

Examples

```
.LOCATE 2↵

.LOC 0↵
```

LOGIN Command

Function

The LOGIN command is used to gain access to the system. LOGIN loads a Monitor Support program that accepts your project and programmer numbers followed by your password. To LOGIN, type these numbers, then wait for the system to print PASSWORD:. Then type your password, which is not echoed. The numbers and password that you type must match those stored in the system accounting file (SYS:ACCT.SYS). LOGIN waits two minutes for your response to PASSWORD: before printing an error message, killing your job, and asking you to start over.

Command Format

```
LOGIN "proj,prog" /switch or
LOGIN [proj,prog] /switchl/switch2 ...
```

The notation proj,prog stands for your project-programmer numbers. These may be separated by either a comma or a slash. If you use a slash, you do not receive the daily message unless the date of the file containing it (NOTICE.TXT) is later than the last date that you logged in. If it is later, the system prints the daily message on the first login only. If you use a comma, you receive the message regardless of your last login time. The notation [path spec] represents your default path for disk I/O when you are logged in. If it is the same as your proj,prog numbers, you may omit it. One example of a path spec is [35,3572,FOOBAR,STUFF]. You may type the proj,prog argument on the same line as the LOGIN command; if, however, you type a carriage return after LOGIN, you may type the argument on the next line after the system prints the number sign (#).

The following switches can be used to modify the command string. These switches may also be in SWITCH.INI.

Most parameters set by these switches can be changed after login by the appropriate monitor commands. (See for example the SET commands and R SETSRC.)

Switch	Meaning
/ALTMODE	converts the ALTmode codes of 175 and 176 (sent by some terminals to the system) to 033, the ASCII standard code for the ESCape character. (Normally, however, your terminal is initially set to send the 033 code.)
/NOALTMODE	restores the individual identity of the codes 175 and 176.
/BLANK	restores your terminal's characteristic (usually part of its initial setting) of printing carriage return/line feeds and form feeds at specified points in a program (before and after a DIR list, for example).

Switch	Meaning
/NOBLANK	suppresses blank lines (consecutive carriage return/line feeds after the first) and outputs form feeds and vertical tabs as two carriage return/line feeds. If your terminal is a video type, you can use this switch to increase the screen's capacity for output.
/CORE:nP	tells the system that the job may use a maximum of n 512-word pages.
/CORE:nK	tells the system that the job may use a maximum of n 1024-word blocks.
/CRLF	restores the execution of a free carriage return/line feed at the terminal's right margin. Normally, the /CRLF condition is an initial setting.
/NOCRLF	suppresses the execution of a carriage return/line feed at the terminal's right margin.
/DEBREAK	tells the system that your terminal has a feature that allows the computer to lock the keyboard. (This feature exists only in Model 2741 terminals.)
/NODEBREAK	turns off the DEBREAK switch.
/DSKFUL:ERROR	causes a full disk to stop output and passes an error condition to the program. Most programs respond to this condition by issuing an error message and returning the job to monitor level without giving you a chance to continue. The default setting is ERROR unless specified as PAUSE.
/DSKFUL:PAUSE	causes a full disk to stop output and suspends execution of the program. The system prints an error message on your terminal and returns control of the job to the monitor. To request assistance from the operator, you may issue the SEND command. To resume execution, however, you may issue a CONTINUE. The program will resume as long as you do not issue a command that destroys the interrupted program's core image. The program will be stopped again, however, if the problem of insufficient disk space, or of an exceeded quota, has not been corrected in the interim. The default setting is ERROR unless specified as PAUSE.

Switch — Meaning

/DSKPRI:n

allows a privileged user to set the priority for his job's disk operations (data transfers and head positionings). Because the normal timesharing priority is 0, and permissible values range from 3 to +3, the priority n can be higher or lower than standard. The specified priority applies to all disk I/O channels opened whose priority is not explicitly set with a DISK monitor call. (See DECsystem-10 Monitor Calls Manual.) The priority specified in DSKPRI remains in effect until 1) a SET DSKPRI command with a different priority is issued, 2) a KJOB command is issued, or 3) your program overrides the /DSKPRI switch by issuing a DISK monitor call with a different priority. Restrictions are as follows: the privileges required for using this switch are determined by bits 1 and 2 of the privilege word, .GTPRV; these two bits specify an octal number from 0-3; you are always allowed a 0 priority.

/ECHO

restores the normal echoing of each character typed in. Most terminals on the DECsystem-10 are used in full-duplex mode. In this mode, the monitor normally transmits to the terminal those characters that are to be printed; this process is called "echoing." If your terminal prints ^C when you type it, but none of the other normally printing characters when you type them, you can use this switch to establish normal echoing. (The initial setting is determined by each installation when the monitor is built and its characteristics set.)

/NOECHO

suppresses monitor echoing of input characters. Local copy terminals (i.e., terminals that automatically print each character as you type it) do not require normal monitor echoing, because the echoing would cause each character typed to be printed twice. This switch, therefore, should be used to stop double printing of each character typed on a local copy terminal. (The initial setting is determined by each installation when its monitor is built.)

/FILL:n

tells the system that the filler class n is assigned to your terminal. Many different types of terminals are supported by the DECsystem-10 operating system. Some terminals require one or more filler characters to be sent following certain control characters such as LF (line feed) and HT

Switch Meaning

(horizontal tab). The filler characters are CR (215 octal for even parity) for carriage return characters (CR, 215 octal) and DEL (RUBOUT, 377 octal) for all other characters. No fillers are supplied for image mode output. For the number of fillers sent for each character and filler class, refer to the table included in SET TTY FILL and NO FILL descriptions in this manual. (The initial setting is determined by each installation when its monitor is built.)

/FORM tells the system that your terminal has hardware FORM (PAGE) and VT (vertical tab) characters, thereby allowing you to use them. (The initial setting is determined by each installation when its monitor is built.)

/NOFORM tells the system that your terminal does not have hardware FORM and VT characters, thereby causing the monitor to simulate them by sending eight line feeds for a FORM and four for a VT. (The initial setting is determined by each installation when its monitor is built.)

/GAG allows your terminal to receive SEND messages, but only at monitor level. Normally, the /GAG condition is an initial setting.

/NOGAG allows your terminal to receive SEND messages even when it is not at monitor level.

/GUIDELINE specifies that the numeric value cited in the /PHYSICAL switch is a guideline. If not issued, /GUIDELINE is assumed; it is the default setting for /PHYSICAL.

/LC tells the system that your terminal has lowercase characters, and therefore prevents their translation to uppercase.

/NOLC causes the monitor to translate lowercase characters to uppercase as they are received. Some terminals have only uppercase characters, while others have both upperacase and lowercase. Frequently, it is convenient to have a terminal with both cases simulate the behavior of one with uppercase only. /NOLC causes this simulation. The monitor's echo always matches the case of the characters after translation. By looking at the printout, you can determine what translation the monitor has performed. Normally, the /NOLC condition is an initial setting.

Switch	Meaning
/LIB:[proj,prog]	sets your job's library directory to the UFD [proj,prog] and adds it to your DSK specification. Consequently, a library area is set up to be searched. (See SETSRC, in this chapter.)
/NOLIB	removes the library directory (LIB:) from your DSK specification.
/LIMIT	specifies that the numeric value cited in the /PHYSICAL switch is a limit rather than a guideline. If /LIMIT is not issued, the system assumes that the numeric value given for the /PHYSICAL switch is a guideline.
/LOCATE:n	logically establishes your job at a specified station. When the job is initiated, your logical station corresponds to your physical station. You will need this switch, therefore, only in changing your logical station. When issuing the switch, type n to be the desired station number.
/NAME:name or /NAME:"name"	associate the given name with your job. This name will appear on QUEUE'd output listings and on other information output by the system. The name should be enclosed in quotes if it contains punctuation characters. (This switch most commonly applies to BATCON version 12 and SPRINT version 2.)
/NEW	tells the system to add the NEW directory to your SYS specification. As a result of this switch, the system, in conducting a search for any program or file, will first check the experimental directory [1,5]. The default setting is /NONEW.
/NONEW	removes the [1,5] directory (NEW) from your SYS specification. It is the default setting for /NEW.
/NOTE:filespec.	prints the specified file, using the NOTICE.TXT logic. That is, it must be followed by the file name and extension. It is useful for printing a project notice that is being kept in a library area. For example, NOTE:*.TXT[33,437] will type all files with the extension .TXT. If the login project number is separated from the programmer number by a slash (/) instead of a comma (,), the files will be typed only if they were created after the last login.

Switch   Meaning

/PAGE   enables you to temporarily suspend system typeout without losing it. /PAGE causes this condition by turning on two features: the XOFF key (^S), which suspends the typeout, and the XON key (^Q), which restores it. These keys are neither echoed nor sent to your program. This switch is useful if you are using a video terminal and wish to read a page of text before it moves off the screen. Note that this switch preempts the use of ^S and ^Q for reading paper tape. (See /TAPE.)

/NOPAGE   disables the typeout control functions of the XOFF and XON keys. The current interpretation of these keys depends on the last /TAPE issued. Normally, the /NOPAGE condition is an initial setting.

/PHYSICAL:nP   specifies the maximum current physical page limit (CPPL; see Chapter 1 of the Monitor Commands Manual). Because P stands for "pages," the notation n represents the maximum number of pages. 1P, therefore, would set a limit of one 512-word page. This switch is used only on virtual memory systems. The value that it specifies will be either a PHYSICAL GUIDELINE or a PHYSICAL LIMIT, depending on whether you type /PHYSICAL:nP/GUIDELINE or /PHYSICAL:nP/LIMIT. If you type neither, /GUIDELINE is assumed.

/PHYSICAL:nK   specifies the number of blocks (K) used as the maximum current physical page limit; 1K would set a limit of one 1024-word block. This switch is used only on virtual memory systems. The value that it specifies will be either a PHYSICAL GUIDELINE or a PHYSICAL LIMIT, depending on whether you type /PHYSICAL:nP/GUIDELINE or /PHYSICAL:nP/LIMIT. If you type neither, /GUIDELINE is assumed.

/QUOTA:str or
/QUOTA:(str1,...)   recomputes disk usage on all specified file structures. If none is specified, usage is recomputed on all file structures in your search list.

/RCVSPEED:n   sets to n baud the speed at which the system receives.

/RUN:filespec   will run the specified program unless another is designated by the system manager.

/SCAN   sets the scan switch for the directory path. (See SETSRC, in this chapter.)

Switch	Meaning
/NOSCAN	cancels the scan switch for the directory path.
/SETTTY	tells the system that your directory has a SWITCH.INI file that contains LOGIN followed by one or more switches that designate characteristics of your terminal only (for example: LOGIN /FORM /NOBLANK /SETTTY WIDTH:36). /SETTTY, therefore, causes the settings in this file to supersede their monitor counterparts. The /SETTTY condition is the default setting.
/NOSETTTY	tells the system not to change any of your terminal's characteristics as designated in the monitor or in any SET TTY commands you have have issued before logging in.
/SPEED:n	sets to n baud the speed at which the system sends and receives. (You can adjust your terminal's speed by setting its baud knob or button.)
/SPOOL:dev or /SPOOL:(devl,...) 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 typed 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 typed on his terminal.
/NOSTR	inhibits the printing of SYS:STR.TXT
/SYS	adds the SYS specification to your DSK specification. Consequently, if a file is not found in the directories on 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.
/NOSYS	removes the SYS specification from your DSK specification.

Switch Meaning

/TABS tells the system that your terminal has hardware TAB stops every eight columns, thereby allowing you to use this feature. (The initial setting is determined by each installation when its monitor is built.)

/NOTABS tells the system that your terminal does not have hardware TAB stops, thereby causing the monitor to simulate them by sending the necessary amount of SPACE characters.

/TAPE allows your terminal to start and stop reading paper tape. /TAPE establishes this function by enabling the XON key (^Q) to start the reading process, and the XOFF key (^S) to stop it. (See Chapter 5 of DECsystem-10 Monitor Calls.)

/NOTAPE disables the special paper tape function of the XON and XOFF keys. These keys may have a PAGE function independent of /TAPE or /NOTAPE. Normally, the /NOTAPE condition is an initial setting.

/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 may be continued by typing CONT, but unless the time is reset via the SET TIME command, no time limit will be in effect. A batch job cannot be continued.

/VIRTUAL:n specifies the current virtual page limit, as represented by n. (For a description of CVPL, see Chapter 1 of the Monitor Commands Manual.) In /VIRTUAL:nK and :nP, K represents a block, and P a page; 1K equals 1024 words, and 1P, 512 words. If you type neither letter, K is assumed. K may be specified within the range 1 to 512P. If /VIRTUAL:n is issued with a 0 argument, the value of CVPL, as set by the system administrator, is used.

/WATCH:(arg(1), arg(2),...arg(n)) sets the system to print the specified file incremental job statistics automatically. Thus, it provides you with a tool for measuring the performance of your programs. For example, you may type a line that reads /WATCH:READ or /WATCH:RUN or /WATCH:WAIT or /WATCH:DAY or /WATCH:VERSION or /WATCH:NONE or /WATCH:MTA or /WATCH followed by any combination of these arguments in parentheses and separated by commas. To include all these arguments, type /WATCH:ALL.

LOGIN Command (Cont.)

Switch	Meaning
/NOWATCH	inhibits the printing of incremental job statistics.
/WIDTH	sets to n the carriage width (the point at which a free carriage return is inserted). The range of n is 17 (two TAB stops) to 200 decimal. Normally, n is initially set to 72.
/XMTSPEED:n	sets to n baud the speed at which the system sends.

Characteristics

The LOGIN command:

Returns your terminal to monitor mode or starts a program running if specified in

ACCT.SYS entry for proj,prog.

Runs the LOGIN program.

Example

The following is the procedure used to gain access to the system.

```
.LOGIN
JOB 29      SCLIKA KL10 SYS#1026 TTY250
#27,4662
```

LOGIN types your assigned job number (job number 29), followed by monitor name, version number, and console line number. If you do not type your project-programmer number on the same line as the LOGIN command, LOGIN outputs a number sign indicating that you should type your project-programmer number.

```
Password:
```

System requests you to type your password. You type password, followed by a carriage return (refer to Section 1.4.2.1). To maintain password security, the monitor does not echo the password. On terminals with local-copy (refer to DECsystem-10 Monitor Calls), a mask is typed to make the password unreadable.

```
1459    17-Aug-77        Wed
```

If your entries are correct, the system responds with time, date, day of the week, the message of the day (if any), and a period, indicating readiness to accept another command.

MAKE Command[1]

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 and supersedes the file, if you continue in TECO. If you type two CTRL/Cs to leave TECO, the system will not destroy the file. (Refer to the TECO manual in the DECsystem-10 Software Notebooks.)

Command Format

```
MAKE dev:file.ext[project,programmer]
```

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.

[project,programmer] is the directory area in which the system creates the file. If you omit the project-programmer number argument- the system assumes your default directory area (i.e., your project-programmer number). Note that the default directory may be an SFD or some other UFD.

You can pass switches to TECO when you precede the switch with a slash in the MAKE command string. When COMPIL interprets the command string, it passes the switch to TECO.

Characteristics

The MAKE command:

Places your terminal in user mode.
Runs the TECO program, destroying your core image.

Example

```
.MAKE TEST3.MIC
```

[1] This command runs the COMPIL program, which interprets the commands before running TECO.

MOUNT Command

Function

The MOUNT command allows you to request assignment of a device via the operator. This command is similar to the ASSIGN command, but, whereas the ASSIGN command operates without operator interaction, the MOUNT command requests operator interaction when necessary. For example, if a batch job requests a DECtape drive and all drives are in use, then the operator can free one for you, if you wish. With this command you can request devices from the restricted pool of devices, which you cannot do with the ASSIGN command.

The MOUNT command gives the operator control over the assignment of devices on the system. When you request a device via this command, the operator has the option of selecting a specific unit (e.g., DTA5), or deferring your request, or canceling your request completely. For example, all units of this type are in use and the operator does not want to free one for you. The operator may also mount the medium for the requested unit if it is sufficiently identified (e.g., a deck of cards in the card reader or an identified DECtape on a specific drive).

When you issue the MOUNT command to gain access to a file structure, you can specify a particular drive, and, if desired, place the file structure name at the end of your job's search list and wait for completion of operator action. Each file structure can have an administrative file, QUOTA.-SYS, which contains a list of quotas for all users allowed to access the structure. (Quotas for the structures in your default search list are contained in the file SYS:AUXACC.SYS.) When the operator mounts a file structure, the system creates a UFD for you if you have an entry in the file structure's QUOTA.SYS or in SYS:AUXACC.SYS.

The MOUNT command runs the UMOUNT program in your core area. UMOUNT scans the command string and completes as much of the command as possible without operator intervention. When operator intervention is required, UMOUNT queues a request to the OMOUNT program by writing a command file in the [3,3] disk area. OMOUNT examines these command files and interacts with the operator. When you delete the command file, the operator action has been completed. UMOUNT waits for this completion of operator action unless you have specified /NOWAIT or you type two CTRL/Cs. When you type two CTRL/Cs, you do not receive a message of confirmation, but you can later use the /CHECK switch to see if your request is still pending. (See the examples.)

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

Command Format

MOUNT dev:logical-name/switches(drives)

dev: is one of the following:

1. A physical or generic device name (e.g., DTA3, CDR0, MTA)

2. A logical name you previously associated with a physical device by either a MOUNT or an ASSIGN command

3. A file structure name (one that is already mounted or one whose name appears on STRLST.SYS)

4. A spooled device name (e.g., LPT, PTP), assuming these devices are spooled

5. NUL:. This argument is required except when specifying one of the following switches: /CHECK, /HELP, or /WAIT. If you issue a MOUNT command for one drive (e.g., MTA1) and the operator selects another drive for you (e.g., MTA2), then the system gives the logical name of MTA1 to the physical drive (MTA2) so that batch control files that reference MTA1 will work correctly.

Logical-name is any SIXBIT name. The logical name may be a previously assigned name. If your request in which this name appeared has been processed, then MOUNT transfers the name from the previously assigned device to the device in this MOUNT command. If the previous request has not been processed at the time you specify the logical name again, then MOUNT creates a unique logical name of the form "Mnnnnn" instead of transferring the name from one physical device to another. It is recommended that you supply a logical name with any non-disk MOUNT request, because when you use generic device names, the logical name is the only way for you to know what physical device has been assigned by the operator. If you omit this field, UMOUNT either creates a unique logical name of the form "Mnnnnn" or it uses the physical device name. UMOUNT creates a unique logical name if you specify a generic device or if you specify a physical device that already has its name assigned as a physical or logical name by your job. UMOUNT uses the physical device name as a logical name in cases where you specify a specific device and the name is not already assigned as either a physical or logical name.

You can use switches to further describe the device to be mounted or the requested action. These switches are optional unless otherwise indicated in the individual switch descriptions. The system processes the switches in the order you specify. Therefore, if there are any conflicting switches, the system uses the last one scanned in the command line. (An exception tot his is the use of the /WAIT and /NOWAIT switches within a single command string. If you specify both switches together, a fatal error results.) You may abbreviate the switch names to as few characters as are necessary to make the switch unique.

/ACTIVE — Mount the disk in your active search list. This means that the structure becomes part of generic DSK: and the system automatically searches it. (Refer to the SETSRC command.) /ACTIVE is the default condition.

/PASSIVE — Mount the disk in your passive search list. The disk is not considered part of generic DSK: and is not automatically searched; rather, it is merely available for use. (Refer to the SETSRC command.)

/CHECK — Check and list your job's pending MOUNT requests. This switch is intended to be used by itself with the MOUNT command, and if a device specification or other switches appear in the command string with the /CHECK switch, the system ignores them.

NOTE

> /C is a valid abbreviation of /CHECK. Therefore, if /CREATE is desired, you must type at least "/CR".

/CREATE — Permit creation of files on this structure, default condition. Specification of this switch implies that /ACTIVE is in effect. (Refer to the SETSRC command.)

/NOCREATE — Prohibit creation of files on this structure when DSK: is specified or implied. Files may be created on this structure if you explicitly use its name in the command line. Specification of this switch implies that /ACTIVE is in effect. (Refer to the SETSRC command.)

/HELP — Type out a brief description of the MOUNT command. This switch is intended to be used by itself with the MOUNT command, and if a device specification or other switches appear in the command string with the /HELP switch, they are ignored.

/MULTI — Allow multiple access, disk only, opposite of /SINGLE. /MULTI is the default condition.

/PAUSE — Notify you (to give you an opportunity to cancel the request) before queuing the mount request to the operator.

/REELID:name — A reel identification to be used when requesting a magnetic tape to be mounted. The argument, "name," is a 6-character ASCII string that is used to uniquely identify a reel of magnetic tape. This identification is included in any system-generated error or status reports about the drive while the reel is mounted. This switch is required when requesting magnetic tapes to be mounted.

/RONLY — Read only, same as /WLOCK.

/SHOVE — Force the mount request to the operator even if SCHED 400 is set.

/NOSHOVE — Do not force the mount request to the operator even if SCHED 400 is set, default condition.

/SINGLE — Prohibit access to this structure by other users. You must be in the same project as the owner of the pack to be allowed single access. File protection is enforced for you. This switch is applicable only to disk mount requests.

/VID:text

A visual identification describing the volume (DECtape, card deck, etc.) to be passed to the operator. This switch is required in MOUNT requests for DECtapes and for all input-only devices.

The "text" can be up to 50 characters in length. For DECtapes, the text is the DECtape number, e.g., /VID:1326. If the tape is not numbered or if you do not wish to have the number checked, you must describe the tape by enclosing the "text" in single or double quotation marks. For example.

```
MOUNT DTA:/VID:"FRIDAY'S FAILSAFE FOR
[13,2652]"
```

For input-only devices, the text should be a description of the input. For example.

```
MOUNT CDR:/VID:"PLEASE LOAD THE BLUE
CARD-DECK:'COBOL SOURCE'"
```

Unquoted strings may contain only alphanumeric characters, periods, and hyphens. In quoted strings, the quotation marks are included in the 50 characters.

/WAIT

Wait before continuing with your job when operator intervention is required. When this switch is used in a command with a device specification and other switches, your job will wait for the request to be completed. When this switch is used alone (MOUNT/WAIT), your job will not continue until all of your pending requests have been processed. This format is useful only if a MOUNT dev:/NOWAIT was done previously or if you have typed two CTRL/Cs. In either case, the message "Waiting...Type 2 CTRL-Cs to Exit" is typed on your terminal. /WAIT is the default condition.

NOTE

/W is a valid abbreviation of /WAIT.

/NOWAIT

Do not wait if the operator intervention is needed to process your pending requests. This switch is particularly advantageous when running batch jobs. In timesharing jobs the use of the /NOWAIT switch merely saves you from typing two CTRL/Cs (^C^C).

/WENABL

Permit writing for your job. This is the default condition for disk.

MOUNT Command (Cont.)

/WLOCK — Prohibit writing for your job. Your job may not write on this device or file structure and the monitor will not update BAT blocks or the access date. If you specify /SINGLE , the operator may set hardware write-lock to ensure that nothing is written. In addition, the write-status of all tapes is verified so you must specify the correct setting. Hardware write-status is not checked for disk and a [1,2] job may write on a disk mounted with /WLOCK, so care should be taken if you are [1,2] and do not wish to write on the disk. /WLOCK is the default for magnetic tape and DECtape.

/WRITE:NO — Same as /WLOCK. Permit writing. This is the default condition for disks.

/WRITE:YES — Same as /WENABL. Prohibit writing. This is the default condition for tapes.

(drives) are the physical drives on which the units are to be mounted. A drive argument may be used only when mounting file structures. The drives must be in the logical unit order within the file structure. Drive names are separated by commas. Leading and embedded drives that are not specified must be represented by null names (,,DPS3). Unspecified trailing drives may be omitted. Drive names are as follows:

Blank, null - unspecified, UMOUNT finds one of proper type.

Two letters - controller class (e.g., DP).

Three letters - specific controller (e.g., DPA). UMOUNT finds drive on that controller.

Three letters and one or two digits - specific drive (e.g., DPA0, DPA1).

By specifying a drive list, you may only suggest that the packs be mounted on specific drive,s or controllers. If you do not specify a drive, the system finds an available drive of the proper type. if you specify a drive incorrectly for the named structure, or the drive is unavailable, you are informed and the requests are then queued.

Characteristics

The MOUNT command:

Places your terminal in user mode.
Runs the UMOUNT program, destroying your core image.

Examples

```
.MOUNT MTA FAILSA/REELID:456/VID:"THURSDAY'S FAILSA TAPE FOR DSKA"↲
REQUEST QUEUED
WAITING...2 ^C'S TO EXIT
^C
^C

.
```

This MOUNT command requests that the operator mount the magnetic tape identified as "Thursday's FAILSA for DSKA." This identification is accomplished either by a label affixed to the reel or canister or by an understanding between the operator and the requesting user. The /REELID switch specifies that the tape is to be identified as reel number 456 in all error reports. In order to associate the error reports with the reel on which the errors occurred, it is recommended that a label bearing the reel identification be attached to the tape reel. This command also specifies that the logical name FAILSA is to be associated with the physical drive on which the reel is mounted.

```
.MOUNT MTA1/REELID:123/VID"PAYROLL MASTER"↵
REQUEST QUEUED
WAITING...2 ^CS TO EXIT
MTA MOUNTED, MTB101 USED

.SYSTAT↵
24      10,4072 TTY11   SYSTAT  12+SPY  RN                   23 $
$ MEANS EXECUTE ONLY

BUSY DEVICES:
DEVICE  JOB     WHY     LOGICAL

MTB101  24      AS      MTA101

.
```

This MOUNT command asks the operator to mount a magnetic tape with a REELID of 123 and the external label "PAYROLL MASTER" on MTA1. The operator was unable to use MTA1 so he selected MTB1 instead. However, OMOUNT assigned the logical name MTA1 to the physical device MTB1. This feature is especially useful in batch control files and allows you to continue to refer to the tape as MTA1.

```
.MOUNT BLKC/NOWAIT↵
REQUEST QUEUED TO MOUNT UNITS
```

This MOUNT command asks the operator to mount the file structure BLKC:. You want to return to monitor command level after the request is queued, without waiting for the operator to service the request.

```
.MOUNT DTA/VID:1234↵
REQUEST QUEUED
WAITING ...2 ^C'S TO EXIT
^C
^C
```

Then you ask the operator to mount the DECtape labeled both magnetically and externally, 1234. You did not specify /NOWAIT so you received the "Waiting...2 ^Cs to Exit" message. However, you decided not to wait and returned to monitor command level by typing two CTRL/Cs.

```
.MOUNT DTA/VID:5678↵
REQUEST QUEUED
WAITING...2 ^C'S TO EXIT
DTA MOUNTED, DTA101 USED.
```

Next, you request that the operator mount the DECtape labeled 5678. This time, you wait until the DECtape is mounted before continuing with your job. You are notified that the DECtape is mounted on DTA1.

```
.MOUNT/CHECK↵
1. M JOB5 TTY105 10,4072 1 MOUNT DTA M44321 /V:1234 /WL /AFTER:14-MAR
-75:17:20
1 COMMAND IN QUEUE
```

The response from the MOUNT/CHECK command indicates which of your requests have not been processed. Because BLKC is not listed in the queue, that request has been serviced. You can type "RESOURCES" to see if the operator mounted the file structure. The mount request for DECtape 1234 is still pending. /AFTER indicates that the operator has deferred servicing this request until 17:20 on 14-Mar-77. The last line indicates that the total number of MOUNT requests pending for all users is one.

```
.MOUNT/WAIT↵
WAITING...2 ^C'S TO EXIT
```

You now decided to wait until all your pending requests have been processed. The line "Waiting...2 ^Cs to Exit" indicates that UMOUNT is waiting.

```
.MOUNT/CHECK↵
NONE PENDING FOR YOUR JOB
1 COMMAND IN QUEUE
.
```

Finally, you verified that all your requests have been processed. Note that some other user has issued a MOUNT command since you last issued the MOUNT/CHECK.

Function

The MTCOPY program copies entrie contents of one magnetic tape to another magnetic tape, or copies selected files from one magnetic tape to another magnetic tape.

Command Format

```
R MTCOPY
*output MTA:/output switches = input MTA:/input switches
```

The file switches are:

/2	200 bits/inch
/5	556 bits/inch
/8	800 bits/inch
/A:n	Advance n files before operation
/B:n	Backspace n files before operation
/E	Even parity
/I	Industry-compatible mode
/R	Rewind before and after operation (default)
/U	Unload after operation

The general switches are:

/C:n	Copy n files (input only)
/C	Copy to double EOF
/G	Proceed on errors
/H	Help
/N	Suppress automatic rewinds
/V:n	Verify n files
/V	Verify to double EOF

Neither /C nor /V defaults to /C/V. A count on either /C or /V applies to both.

The system preserves density across commands until you specify that the tape is to be unloaded or rewound. /A, /B, and /N suppress automatic /R. /R without /A or /B restores automatic rewinds. Devices need not be reset for new commands.

Spaces and tabs are ignored. Comments are preceded by a semicolon; switches may be included in parentheses.

Characteristics

The R MTCOPY command:

Places your terminal in user mode.
Runs the MTCOPY program, destroying your core image.

Examples

.R MTCOPY↵	Run MTCOPY.
*MTA1:C/V/2=MTA2:↵	Copy and verfiy a 200 bits/inch magtape from MTA1: to MTA2:.
*MTA2:/A:2/C:3/V:3- =MTA3:↵	Copy and verify files 3 to 5 from MTA2: to file 4 on MTA3:.
*MTA0:A:4=MTA0:↵	Position MTA0: to the beginning of the fourth file.

NODE Command

Function

The NODE command has four distinct functions, determined by the number of arguments given in the command string.

Number of Arguments	Function
0	Type out the node-id and related information for each known node in the network.
1	Type out the node information of the specified node.
2	Assign the designated I/O device on the specified node. The logical device name is the same as the physical device name.
3	Assign the designated I/O device on the specified node. The logical device name is the one given in the command string.

Command Format

NODE node-id physical-device-name logical-device-name

node-id is the node identifier of the node on which the device is to be assigned.

physical-device-name is any physical device name followed by a 1- to 3-digit number representing a unit number. If you omit n, the monitor attempts to assign any device of the designated type or any file structure name.

logical-device-name is a logical name of up to six characters you assign. This argument is optional and if omitted, the logical name will be the same as the physical name. Except for disk devices, you can assign only one logical name to a physical device. Subsequent NODE commands to all devices, except disk devices, replace the old logical name with the new one. You can disassociate all logical names from all devices by issuing the DEASSIGN command.

Characteristics

The NODE command:

Leaves your terminal in monitor mode.

Restrictions

You must use a space, tab, or underscore to separate the node-id and the physical device name. You must use a space or a tab to separate the physical and logical device names.

Examples

```
.NODE
NODE    KL1026(26)      SCLIZA KL10 SYS#1026 07-24-77
        MCR[1] CDR[1] LPT[3] PTR[1] PTP[1] MTA[9] DTA[8] TSK[16]
NODE    CTCH22(22) DN82 V11(24) 07-JUN-77
        TTY[33] CDR[1] LPT[1]

.
```

NSAVE Command

Function

The NSAVE command writes out a core image of your core area on the specified device. (Note that files on DECtape cannot be NSAVED.) If the file to be saved is in .EXE format, only one file will be written instead of the standard high and low files. When the program is subsequently loaded by a GET, R, or RUN command, it will be nonsharable.

Command Format

NSAVE dev:file.extension[proj,prog]core

Arguments and defaults are the same as in the OSAVE command.

Characteristics

The NSAVE command:

Leaves your terminal in monitor mode.
Requires core.
Does not operate when a device is currently transmitting data.

Examples

```
.NSAVE TEST↵
TEST SAVED
```

NSSAVE Command

Function

The NSSAVE command writes out a core image of your core area on the specified device. (Note that files on DECtape cannot be NSSAVEd.) If the core image to be saved is in the new .EXE format, only one file will be written instead of the standard high and low files. When the program is subsequently loaded by a GET, R, or RUN command, it will be sharable.

Command Format

```
NSSAVE dev:file.extension[proj,prog]core
```

Arguments and defaults are the same as in the OSAVE command.

Characteristics

The NSSAVE command:

Leaves your terminal in monitor mode.
Requires core.
Does not operate when a device is currently transmitting data.

Example

```
.NSSAVE TEST↵
TEST SAVED
```

OPSER Program

Function

The OPSER program facilitates multiple job control from a single terminal by allowing you to run up to 14 jobs, called subjobs, from your terminal. The OPSER program acts as the supervisor of the various subjobs by allowing monitor level or user level commands to be passed to all of the subjobs or to selected subjobs. Output from the various subjobs may be retrieved by OPSER.

The subjobs of OPSER run on pseudo-TTYs (refer to DECsystem-10 Monitor Calls) and all initializations of the pseudo-TTYs are performed by OPSER. You need only to provide the subjob name and either an OPSER-provided subjob number or a user assigned name. System programs, as well as programs you write that require a dedicated terminal can be run as subjobs of OPSER. By running jobs on pseudo-TTYs, OPSER is able to maintain an I/O link between you and the running jobs. In addition, the output from the various subjobs is concentrated on one terminal instead of many, as was the case when each program required its own terminal.

The OPSER program is primarily intended for use by the system operator in controlling the separate components of the multiprogram batch facility. However, all users can employ this program for their own purposes. (Refer to the OPSER specification in the DECsystem-10 Software Notebooks and Appendix A of the Operator's Guide for information on OPSER.)

Command Format

R OPSER

OPSER signifies its readiness to process commands by typing an asterisk if no subjobs are in use or subjobs are in a wait state for an operator action. OPSER responds with an exclamation point when a subjob is running. Commands may be entered whenever OPSER is operating. Each command is preceded by a colon and must be typed to sufficient length to make it unique.

OPSER Commands

Command — Meaning

:AUTO/hh:mm

Process the specified file as an automatic startup file. The execution of the file is terminated by an end-of-file or by the operator typing on the console. This is the normal way that the standard subjobs are started by the operator. The time argument is optional: when it is given, the AUTO file is run at the specified time. Comments can be included in an AUTO file by preceding the comments with a semicolon. For example,

```
:SLOGIN
:LOGIN A NEW SUBJOB
```

OPSER Program (Cont.)

OPSER Commands (Cont.)

Command	Meaning
:AUTO/ >hhmm	Process the AUTO file at the next occurrence of hhmm.
:AUTO/ >hhmm	Don't process the AUTO file if time is past hhmm.
:CLOSE	Close the disk log file without opening a new one.
:CONTINUE	Continue processing the AUTO file after it has been interrupted by a Control-C. This allows the operator to gain control of a subjob during auto file processing.
:CURRENT	Type name, if defined; otherwise, type the number of the current subjob (the last one typed into). Output from another subjob does not affect current subjob.
:DAYTIME	Obtain the current date and time.
:DEFINE xxx=n	Associate the symbol xxx as the mnemonic for subjob number n. The symbol B is reserved for the subjob running BATCON.
:DEVICE nam log:n	Assign the device with the physical name ,nam, and logical name, log, to subjob n. The logical name is optional, but you must type a null field if the name is omitted, e.g., :DEVICE CDR::3. The command re-ENTER will abort the request.
:ERROR n,m,p	Report only error messages (that is, ignore nonerror messages from subjob n). Message reporting is resumed with the :REVIVE command.
:EXIT	Exit to the monitor if no subjobs are in use; otherwise, give a list of those that are running. This should be used instead of ^C, because EXIT does not return your job to the monitor mode if there are any active subjobs. (See also: MONITOR.)
:FREE	Type the first free subjob number.
:HELP	Type a text which briefly explains the commands.
:JCONT n	Continue the specified stopped job.
:KILL n,m,p	Kill the specified subjobs. This is identical to :KJOB.

OPSER Commands (Cont.)

Command	Meaning
:KJOB n,m,p	Kill the specified subjobs, saving all files.
:KSYS hhmm	Operator-privileged command.[1]
:LOGIN proj,prog	Login a new subjob. If no project-programmer number is typed, assume OPSER's project-programmer number.
:MONITOR	Exit to the monitor, even if subjobs are running. (See also :EXIT.)
:MSGLVL 0	Cause the response to the :WHAT command to include the JOBSTS bits.
:MSGLVL 1	Cause the response to :WHAT command to eliminate the JOBSTS bits (default).
:QUEUE <line>	Initiate the first free subjob and send the typed-in line to the system queue manager.
:RESOURCES	Type the list of the available system resources.
:RESTRICT dev:	Operator-privileged command.[1]
:REVIVE n	Resume normal echoing of output from subjob n (i.e., clear effects of :SILENCE, :TSILENCE, and :ERROR for subjob n).
:SCHED	Type out the schedule bits as set by the operator. Bits are: 0 Regular timesharing 1 No further LOGINs except from CTY. 2 No further LOGINs from remote terminals, and no answering of data sets 4 Batch jobs only 100 Device MOUNTs can be done without operator intervention 200 Unspooling allowed 400 No operator coverage
:SEND line	Simulate the SEND monitor command.
:SET a	Simulate a SET monitor command. Valid SET monitor commands are SET CORMAX, SET CORMIN, SET DATE, SET DAYTIME, SET OPR TTY, SET SCHED, and SET TTY.

[1] See Appendix A of the Operator's Guide.

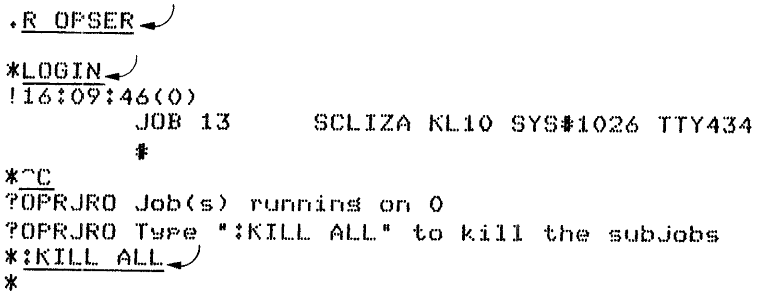
OPSER Commands (Cont.)

Command	Meaning
:SET BATMAX n	Operator-privileged command.[1]
:SET BATMIN n	Operator-privileged command.[1]
:SET LOGMAX n	Operator-privileged command.[1]
:SET RUN	Operator-privileged command.[1]
:SILENCE n	Suppress all output from subjob n.
:SLOGIN proj,prog	LOGIN one subjob but suppress its response. If proj,prog is omitted, OPSER uses its own.
:STOP n	Put the specified subjob in monitor mode. This is equivalent to inputting two CTRL/Cs in interactive mode.
:SYSTAT xx	Run SYSTAT with optional argument xx over the first free subjob.
:TIME	Type out the total running time since the last :TIME command, followed by the integrated product of running time and core size.
:TLOG filespec	Create a disk log file with the specified name. If the file already exists, a message is typed to determine whether the existing file should be superseded. If not, the file is appended to the existing one. Default for filespec is OPSER.LOG.
:TSILENCE n	Suppress all output from subjob n (same as the :SILENCE command) but place entries into the log file.
:TTYTST	Test this terminal by typing all the ASCII characters between octal 40 and 174, inclusive.
:UNRESTRICT dev:	Operator-privileged command.[1]
:WHAT n,m,p	Type the status of the specified subjobs on the terminal. The status includes a SYSTAT with the time, the time of the last input and the last output, a linear listing of the JOBSTS bits depending upon the value of :MSGLVL and the time of the next timed AUTO file.
:WHERE devn:	Type out the station number of the device's physical location.

[1] See Appendix A of the Operator's Guide.

When a subjob number or name is required in a command string, the subjob can be specified in one of four ways. It can be omitted, in which case the last subjob typed into is used; the mnemonic ALL can be used, in which case all active subjobs are implied; decimal number can be used, from zero to the limit OPSER is generated for; or a mnemonic previously assigned to the subjob with the :DEFINE command can be used.

Examples



```
.R OPSER

*LOGIN
!16:09:46(0)
        JOB 13     SCLIZA KL10 SYS#1026 TTY434
        *
*^C
?OPRJRO Job(s) running on 0
?OPRJRO Type ":KILL ALL" to kill the subjobs
*:KILL ALL
*
```

OSAVE Command

Function

The OSAVE command writes out a core image of your core area on the specified device. It saves any user program (two-segment sharable, one-segment nonsharable, or two-segment nonsharable) as one or two files. Later, when the program is loaded by a GET, R, or RUN command, it will be nonsharable. If DDT was loaded with the program, the entire core area is written; if not, the area starting from zero up through break (as specified by .BSA, left half) is written. The save command stores JOBVER in .RBVER when saving .LOW and .SAV files; and it stores JOBHVR in .RBVER when saving .HGH and .SHR files. (Refer to the DECsystem-10 Monitor Calls for a description of the job data area locations referenced by this command.)

You should use the OSAVE command instead of the OSSAVE command when debugging a two-segment program.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (TWO EOFs in a row). When loading, the high segment is always referred to because the low segment file may not exist; the opposite holds for the other .SAV file format.

When running under a virtual memory system, DAEMON will be called on an OSAVE command if any pages are paged out or your job has a non-contiguous core image.

Command Format

```
OSAVE dev:file.ext[proj,prog] core
```

dev is the device on which the core image file is to be written. The default device name is DSK:. In nondisk monitors, the default is the generic name that matches the system device. The colon following the device name is required if a device is specified.

file.ext 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 GET, OSAVE, or OSSAVE command; the last command that ran a program (e.g., DIRECT); or the last SETNAM monitor call.

ext applies to the file name used for the low segment not within the high segment.

[proj,prog] 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.

Note that on KA10-based systems (DECsystem-1040, 1050, 1055), the minimum unit of allocation is 1024 words. Therefore, all arguments are rounded to the nearest multiple of 1024 words (e.g., 3P is treated as 2K on a KA10-based system).

If core is omitted, only the number of blocks required by the core image area (as explained in the RUN command description) is assumed.

Characteristics

The OSAVE command:

Leaves your terminal in monitor mode.
Requires core.
Will not operate while a device transmits data.

1. If a two-segment core image is being saved, the high segment is saved with the file ext. .HGH, and overwrites any previous file similarly named.
2. If TEST.SHR exists when the OSAVE TEST is given, then TEST.SHR is deleted.

Example

```
.OSAVE TEST↵
TEST SAVED
```

OSSAVE Command

Function

The OSSAVE command is the same as the OSAVE command except that the high segment, if present, will be sharable when it is loaded with the GET command. To indicate this sharability, the high segment is written with extension .SHR instead of .HGH. A subsequent GET will cause the high segment to be sharable. Because an error message is not given if the program does not have a high segment, you can use this command to save system programs without having to know which are sharable.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (TWO EOFs in a row).

You should use the OSAVE command instead of the OSSAVE command when debugging the program. This is because a GET command after an OSAVE command does not reinitialize the original high segment from the file after you modify it with the D command or the DDT program.

When running under a virtual memory system, DAEMON will be called on an OSSAVE command if pages are paged out or the core image is non-contiguous.

Command Format

OSSAVE dev:file.ext[proj,prog] core

Arguments and defaults are the same as in the OSAVE command.

Characteristics

The OSSAVE command:

Leaves your terminal in monitor mode.
Requires core.
Will not operate while a device transmits data.

Example

```
.OSSAVE TEST↵
TEST SAVED
```

PJOB Command

Function

The PJOB command causes the monitor to respond with the following information:

1. Your job number
2. Your user name
3. Your project-programmer number
4. Your terminal number

This command is in the KAFULL, KIFULL and KLFULL monitors, under the feature test switch FTPJOB.

Command Format

PJOB

Characteristics

The PJOB command:

Leaves your terminal in monitor mode.

Example

```
.PJOB
JOB 49   USER CUSTER,L.   [27,4072]   TTY240

.
```

PLEASE Command

Function

The PLEASE command allows you non-conflicting two-way communication with the designated station operator.

Command Format

PLEASE dev:prog! text

dev is any terminal not assigned to a job (i.e., is not a job's controlling terminal) with which you wish to communicate, including:

1. TTYn: directs the text to a specific terminal unit. The default is TTY0.
2. OPRnn: directs the text to the operator's terminal at node nn.
3. (null argument) directs the text to TTY0 at the central node.

prog! is the name of the system program to be run automatically when the message is completed. This argument may appear before or after the device argument and must be concluded with an exclamation point. If PLEASE is entered at the CCL entry point, it reads file nnnPLS.TMP. This file is sent to the designated device. After the operator terminates the request, the specified program will be run at its CCL point. Neither the dev: nor prog! argument can be used from a batch control file.

text is your message. The argument is required. Characters are not transmitted until the RETURN, vertical tab, or form feed is depressed, at which point the entire line is transmitted.

When you depress the RETURN, vertical tab, or form feed key, a message informing the operator of the caller's node number, project-programmer number (or user name if monitor job tables are available), and text message is printed on dev:. In addition, PLEASE prints time information involved in the transmission. An ESCAPE on CTRL/C on either your terminal or dev: causes communication to terminate and your TTY to be left in monitor mode. Note that when the line terminates with an ESCAPE, the line is typed but the operator response is not waited for. Messages may be typed in both directions without retyping the command.

Characteristics

The PLEASE command:

Places your terminal in user mode until you type an ESCAPE.
Runs a system program except when used with batch.

Restrictions

For batch users, the PLEASE command is trapped by the Batch Controller and only PLEASE text is allowed. It can be used to request operator action while in the batch mode. The line of text can only be one line and may or may not be terminated with an ESCAPE.

Example

```
.PLEASE TELL ME WHEN DTA013 WILL BE FREE↵
OPERATOR HAS BEEN NOTIFIED
IN HALF AN HOUR
THANKS↵
^C

.
```

PLOT Command

Function

The PLOT command places entries in the plotter output queue. This command is equivalent to the following form of the QUEUE command:

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

Command Format

```
PLOT jobname = list of input specifications
```

jobname is the name of your job being entered into the queue. The default is the name of the first file in the request, not the name of the first file given. These differ when the first file given does not yet exist.

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

dev: is any disk file structure to which PLTSPL has access; the default is DSK:.

file.ext are names of the files. The file name is optional. The default for the first file name is *, the default for subsequent files is the last file name used. The extension can be omitted; the default is .PLT.

[directory] is a directory to which you have access; your directory is assumed if none is specified.

You can obtain the listing of entries in the plotter queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., PLOT [40,15]). If no arguments are given with the command (i.e., only the command name is given), the entries for all jobs of all users are output. The asterisk wildcard convention can be used for the input specifications. Switches that aid in constructing the queue entry can appear as part of the input specification. These switches are divided into three categories:

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

2. General - Each switch in this category can appear only once in the command string because they affect the entire request. The switch used can appear anywhere in the command string.

3. File control - Any number of these switches can appear in the command string because this category is specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the file name, it becomes the default for subsequent files. For example, the command string

   ```
   PLOT FILEA,FILEB/DISP:REN,FILEC
   ```

indicates that the DISPOSE switch is only for FILEB. The command string

```
PLOT /DISP:REN FILEA,FILEB,FILEC
```

indicates that the DISPOSE switch is for all three files.

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

(See the QUEUE command for a complete list of switches and their functions.)

Characteristics

The PLOT command:

Leaves your terminal in monitor mode.
Runs the QUEUE program, destroying your core image.
Does not require LOGIN when only QUEUE listings are desired.

Examples

`.PLOT *.PLT/FORMS:PLAIN↵`	Cause all files with the extension .PLT in your area to be plotted. Because these are spooled files (i.e., have the extension .PLT), the files are immediately renamed out of your area, and deleted after plotting. The operator is asked to put PLAIN paper on the plotter.

PRESERVE Command[1]

Function

The PRESERVE command renames the specified files with the standard protection (usually 055 to 057) inclusively ORed with 100. The files are then preserved so that KJOB will not delete them unless you request it to. This command has the same action as the P argument to the KJOB command when individually determining what to do with each file.

Command Format

PRESERVE file.ext,file2.ext,file3.ext,...

> You can use the full wildcard construction for either the file name or the extension. If you specify a protection code, the result of the PRESERVE command is the same as the PROTECT command; that is, the given protection code is used instead of the standard protection inclusively ORed with 100.

Characteristics

The PRESERVE command:

> Leaves your terminal in monitor mode.
> Runs the PIP program, destroying your core image.

Example

```
.PRESERVE TEST.REL↵
FILES RENAMED:
TEST.REL

.PRESERVE PROG1.*↵
FILES RENAMED:
PROG1.REL
PROG1.QOR
PROG1.ALG
PROG1.FOR

.
```

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

Function

The PRINT command places entries into the line printer output queue. This command is equivalent to the following form of the QUEUE command:

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

Command Format

```
PRINT jobname = list of input specifications
```

jobname is the name of your job being entered into the queue. The default is the name of the first file in the request, not the name of the first file given. These differ when the first file given does not yet exist.

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

dev: is any disk file structure to which LPTSPL has access; the default is DSK:.

file.ext are the names of the files. The file name is optional. The default for the first file name is *, the default for subsequent files is the last file name used. The extension can be omitted; the default is .LPT.

[directory] is a directory to which you have access; your directory is assumed if none is specified.

You can obtain the listing of entries in the line printer queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., PRINT [10,157]. If you do not specify any arguments with the command (i.e., you type only the command name), the system prints all entries in the line printer queue for all jobs for all users.

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. These switches are divided into three categories:

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

2. General - Each switch in this category can appear only once in the command string because they affect the entire request. The switch used can appear anywhere in the command string.

3. File control - Any number of these switches can appear in the command string because this category is specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the file name, it becomes the default for subsequent files. For example, the command string

PRINT Command (Cont.)

```
PRINT FILEA, FILEB/DISP:REN, FILEC
```

indicates that the DISPOSE switch is only for FILEB. The command string

```
PRINT /DISP:REN FILEA, FILEB, FILEC
```

indicates that the DISPOSE switch applies to all three files.

(See the QUEUE command for a complete list of switches and their functions.)

Characteristics

The PRINT command:

Leaves your terminal in monitor mode.
Runs the QUEUE program, destroying your core image.
Does not require LOGIN when only queue listings are desired.

Examples

```
.PRINT NOTICE.TXT↵
```

Print the file DSK:NOTICE.TXT.

```
.PRINT SYSTAT.SCM/DISP:RENAME/COPIES:0↵
```

Print two copies of the file DSK:SYSTAT.SCM from your default area. Rename the file out of your area immediately and delete it after printing.

```
.PRINT*.TXT/HEADER:Q/FORMS:2PART↵
```

Print all files in your area which have the extension .TXT. Do not print file headers between the files. Print the files on forms known to the operator as 2PART.

```
.PRINT/SEQ:356/KILL↵
```

Remove the request with sequence number 356 from the LPT queue. This is accepted only if the spooler has not started processing the request.

```
.PRINT LOADER.SAV/OKBINARY/PRINT:SUPPRESS↵
```

Print a file known to be a binary file and suppress all carriage control characters except CR and LF.

```
.PRINT PRGMAC.REL/PRINT:OCTAL↵
```

Print an octal dump of the file PRGMAC.REL.

PROTECT Command[1]

Function

The PROTECT command alters the access protection codes associated with the specified files. The action of this command is similar to the R switch in PIP.

You indicate the access protection of a file by three octal digits. Each digit represents a particular class of user. The first digit represents the owner of the file, the second represents users with the same project number as the owner, and the third represents all other users. Each number in the 3-digit code can be one of the following.

7 No access privileges
6 Execute the file
5 Read and execute the file
4 Append, read and execute the file
3 Update, append, read, and execute the file
2 Write, update, append, read, and execute the file
1 Rename, write, update, append, read, and execute the file
0 Change protection, rename, write, update, append, read, and execute the file.

The standard protection is normally 057, which means the owner has all privileges (0), users in the owner's project can read and execute the file (5), and all other users cannot access the file (7). However, the system standard may be changed by the individual installations. Note that the owner of a file can alter the file's protection regardless of the existing protection code. Therefore, codes 0 and 1 for the owner have an equivalent action. In addition, because the owner can LOOKUP the file to change the protection via a RENAME, codes 7 and 6 are equivalent to code 5 for the owner.

Command Format

```
PROTECT file <nnn>, file2 <nnn>, file3 <nnn>,...
```

You can specify the protection code before the file name, in which case it is the default for subsequent files until you change it. You can use the fill wildcard construction for either the file name or the extension.

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 becomes the default for all succeeding files specified in this command line.

If you omit the protection code, the result of the PROTECT command is the same as the PRESERVE command; that is, the system gives the standard protection code inclusively ORed with 100 to the file.

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

PROTECT Command (Cont.)

Characteristics

The PROTECT command:

Leaves your terminal in monitor mode.
Runs the PIP program, destroying your core image.

Examples

```
.PROTECT TEST.REL <057>↵
FILES RENAMED:
TEST.REL

.PROTECT PROG1.*<123>↵
FILES RENAMED:
PROG1.REL
PROG1.QOR
PROG1.ALG
PROG1.FOR

.
```

Function

The QUEUE command allows you to make entries in several system queues - the input queue for the batch system, and the output spooling queues for the line printer, the card punch, the paper-tape punch, and the plotter. The QUEUE command also provides the means of obtaining listings of the entries in the queues.

Command Formats

1. QUEUE INP: jobname = control file specification, log file specification

   To make an entry in the batch input queue, INP:.

2. QUEUE output queue name: jobname = list of input specifications

   To make an entry in an output spooling queue.

3. QUEUE listing file specifications/LIST = list of queue names

   To obtain a listing of the entries in a queue.

4. The following six commands can be substituted for the various formats of the QUEUE command:

   a. CPUNCH jobname = list of input specifications equivalent to QUEUE CDP: jobname = list

   b. PLOT jobname = list of input specifications equivalent to QUEUE PLT: jobname = list

   c. PRINT jobname = list of input specifications equivalent to QUEUE LPT: jobname = list

   d. SUBMIT jobname = control file name, log file name equivalent to QUEUE INP: jobname = control file, log file

   e. TPUNCH jobname = list of input specifications equivalent to QUEUE PTP: jobname = list

Queue names are taken from the following list:

INP:	(I:)	Batch input queue
LPT:	(L:)	Line printer output queue, default condition
CDP:	(C:)	Card punch output queue
PTP:	(PT:)	Paper-tape punch output queue
PLT:	(PL:)	Plotter output queue

Control file specification is the file specification, plus switches and keyword parameters, for the control file being submitted to the batch input queue. This file can be on any file structure that you have access to; the default is DSK:. The file name is required, but you can omit the extension; the default is .CTL. The asterisk construction is legal for the file name or extension and is the only way you can submit multiple control files to batch with a single QUEUE command.

Log file specification is the file specification for the file that is to be used to record action taken during the execution of the control file. This file can be in any directory on any file structure on which you have privileges to write. The default is the same file structure in which the control file resides. If the file name is missing, the log file is given the same name as the control file. If the extension is omitted, it is .LOG.

Jobname is the name of your job being entered into the queue. The default job name is the name of the first file in the request and not the first file given. These names are different when the first file given does not yet exist.

Input specifications are the file specifications for the disk files to be processed and the various switches and keyword parameters that aid in constructing the queue entry. The files can be on any file structure that the queue processor has access to; the default is DSK:. The files can be in any directory, provided that you have read-access to them; the default is your directory. The file name is optional; the default is * for the first file name. The default for subsequent file names is the last file name used. Note that the asterisk wildcard construction is legal only in the input specifications. You can omit the extension because each queue has a default extension for the files to be processed. These default extensions are:

.CTL -	Batch input queue
.LPT -	Line printer queue
.CDP -	Card punch queue
.PTP -	Paper-tape punch queue
.PLT -	Plotter queue

The listing file specification is the description of the listing file. The default for the listing file destination is TTY unless a file name is specified. In this case, the listing goes to the disk. The default extension for the queue listing is .LSQ. If no queue names are specified, all queues for all the jobs of all users are listed.

You can obtain queue listings for specific project-programmer numbers by following the queue name with the desired project-programmer numbers enclosed in square brackets (e.g., QUEUE/LIST = INP: [27,235], [10,47]). The wildcard construction can be used for the project number or programmer number. If no queue names are specified but a project-programmer number is given, all queues for all jobs of the specified user are listed. For example, QUEUE/LIST = [27,400] lists the entries in all queues for the user [27,400].

Switches - Three categories of switches are provided. The first category contains the switches that define the operation; the second contains the switches that can appear only once because they affect the entire request; the third contains the switches specific to each file. In general, switches that precede the file name become the default for all succeeding files. This is true also for a device name, an extension, or a directory name that precedes a file name.

Queue-Operation Switches - Only one of this type of switch can be placed in a command string, because these switches define the type of queue request. This switch may appear anywhere in the command string.

General Queue Switches - Each of these switches can appear only once in a command string. They affect the entire request, generally in terms of scheduling. These switches can appear anywhere in the command string.

File-Control Switches - These switches affect the individual files in a request and must be adjacent to the file name in the command string. In order to change the defaults for the rest of the files, however, these switches must appear before a file name. For example, the command string

```
QUEUE FILEA,FILEB/DISP:REN,FILEC
```

indicates that the DISPOSE switch will affect only FILEB. The command string

```
QUEUE /DISP:REN FILEA,FILEB,FILEC
```

indicates that the DISPOSE switch applies to all three files.

In the table of switches below, the following conventions have been used:

ALL	- Switches that can appear for both the batch input queue and the output queues.
INPUT	- Switches that can appear only for the batch input queue.
LIST	- Switches that can appear only for the listing file specification.
OUTPUT	- Switches that can appear only for the output queues.

Note that if you omit an argument to a switch, you must also omit the colon preceding the argument. Otherwise, the argument is assumed to be zero and not the default value.

Switch	Category	Queues	Meaning
/AFTER:t	General	ALL	Process the request after the specified time. t is in the form hh:mm (time of day) or +hh:mm (time later than the current time). A colon must be used to separate the hours from the minutes. The resulting AFTER time must be less than the DEADLINE time. If the switch or the value of the switch is omitted, no AFTER constraints are assumed.
/BEFORE:t	General	OUTPUT	Queue only the files with creation dates before time t where t = dd-mmmm-yy: hh:mm. Date must be today or earlier.
/BEGIN:n	File Control	OUTPUT (LPT) INPUT	Start the output on the nth page for LPT requests or on the nth line of the control

QUEUE Command (Cont.)

Switch	Category	Queues	Meaning
			file for INP: requests. The default is to begin output on the first unit.
/CARDS:n	General	INPUT	Use n (decimal) as the maximum number of cards that can be punched by your job. If the switch is omitted, no cards are punched. If the switch is given with no value, 2000 cards is assumed as the maximum.
/COPIES:n	File Control	OUTPUT	Repeat the output the specified number of times (n must be less than 64). The default is one copy. If more than 63 copies are desired, two (or more) requests must be made.
/CORE:n	General	INPUT	Use n (in decimal K) as the maximum amount of core memory that your job can use. If the switch is omitted, a maximum of 25K is assumed; if the value of the switch is omitted, a maximum of 40K is assumed.
/CREATE	Queue Operation	ALL	Make a new entry in the specified queue. This switch is the default for the queue-operation switches.
/DEADLINE:t	General	ALL	Process the request before specified time. t is in the form hh:mm (time of day) or +hh:mm (time later than the current time). A colon must be used to separate the hours from the minutes. The resulting DEADLINE time must be greater than the /AFTER time. If the switch or the value of the switch is omitted, no DEADLINE constraints are assumed.
/DELETE	File Control	ALL	Same as /DISPOSE.
/DEPEND:n	General	INPUT	Specifies the initial value of the dependency count in decimal. When used with /MODIFY, this switch changes the dependency count of another job. If n is a signed

Switch	Category	Queues	Meaning
			number (+ or -), that number is added to or subtracted from the dependent job's count. If n is not a signed number, the dependent job's count is changed to n. If this switch is omitted, no dependency is assumed.
/DISPOSE: DELETE	File Control	ALL	Delete the file after spooling.
/DISPOSE: PRESERVE	File Control	ALL	Save the file after spooling. This is the default for all files except those with extensions of .LST.
/ERBINARY			Type an error message if binary file is included in the queue.
/ERNONE			Type an error message if no files match the wildcard construction.
/ERPROTECTION			Types an error message if a violation of protection fields would be required to process the command.
/F	Queue Operation	LIST	List the entries in the queue, but do not update the queues. Therefore, the list may not be an up-to-date listing of the queues but the listing will be faster than with /LIST.
/FEET:n	General	INPUT	Use n (in decimal) as the maximum number of feet of paper tape that your job can punch. If the switch is omitted, no paper tape is punched. If the value is omitted, the default is 10*B+20 feet, where B is the number of blocks in the request.
/FILE:ASCII	File Control	INPUT	Specify that the INPUT file format is to be interpreted as ASCII text. This is assumed for all files with extensions other than .DAT.
FILE:COBOL	File Control	INPUT	Specify that the INPUT file format is to be interpreted as COBOL SIXBIT text.
/FILE:ELEVEN	File Control	INPUT	Specify that the INPUT file format is to be interpreted as four 8-bit bytes in each 36-bit word. These bytes are

Switch	Category	Queues	Meaning
			arranged as follows: 1st byte is in bits 10-17; 2nd byte, in bits 2-9; 3rd byte, in bits 28-35; 4th byte, in bits 20-27.
/FILE:FORTRAN	File Control	INPUT	Specify that the INPUT file format is to be interpreted as FORTRAN ASCII text (obeys FORTRAN control characters). This is assumed if you specify the extension .DAT explicitly.
/FORMS:a	General	OUTPUT	Place the output on the named forms. The argument to the switch must be six alphanumeric characters. Normal forms (14 x 11) are used if this switch is omitted. Narrow forms are 8 1/2 x 11.
/HEADER: 0 or 1	File Control	OUTPUT	Output block headers at beginning of the file if 1 (default); do not output headers if 0.
/HELP or /HELP:TEXT			Print a message giving the general format of the command string and explain the dialogue that is entered if you need additional help.
/HELP: SWITCHES			Type out a list of switches.
/KILL	Queue Operation	ALL	Remove the specified entry from the specified queue. This switch requires an output specification; it does not default to LPT:*. The /KILL switch can be used for deleting a previously submitted request as long as the request has not been started.
/LIMIT:n	General	OUTPUT	Limit the output to the specified number of pages, cards, feet, or minutes. (One page is about three blocks.)
/LIST	Queue Operation	LIST	List the specified entries in the queue; the default entries are those for queues for all the jobs of all users.
/L:ALL	Queue Operation	LIST	List all data on each queue request.

Switch	Category	Queues	Meaning
/L:DETAIL	Queue Operation	LIST	List detailed job data on each queue request.
/L:FAST	Queue Operation	LIST	Same as /FAST.
/L:FILES	Queue Operation	LIST	List the files in each queue request.
/L:JOBS	Queue Operation	LIST	List jobs in the queue.
/LOG	File Control	OUTPUT (LPT)	Define the file that the spoolers will use to record their output. The default is jobname LOG.
/METERS:n or /METRES:n	General	OUTPUT (PTP)	Limit PTP-limit to n meters.
/MODIFY	Queue Operation	ALL	Alter the specified parameters in the specified jobs; this switch requires that you have access rights to the job. It also requires a queue name; it does not default to the LPT. This switch can be used to modify a previously submitted request as long as the request has not been started.
/NEW	File Control	ALL	Accept request even if file does not yet exist. This is the default for the log file of batch input queue.
/NOHEADER	File Control	OUTPUT	Suppress the block headers at the beginning of the file.
/NOLOG	File Control	OUTPUT	This is not to be the job's log file.
/NONEW	File Control	ALL	This file already exists.
/NONULL	General	OUTPUT	Print an error message on a null request.
/NOOPTION	General	OUTPUT	Suppress the option file SWITCH.INI.
/NOPHYSICAL	File Control	OUTPUT	Allow logical names in a LOOKUP.
/NOREMOVE	File Control	OUTPUT	Do not remove a file from the queue.

QUEUE Command (Cont.)

Switch	Category	Queues	Meaning
/NORESTART	General	INPUT	Job cannot be restarted if your job was stopped because of an error.
/NOSTRS	File Control	OUTPUT	Do not scan each structure for files with the same name.
/NOTE:a	General	OUTPUT (LPT)	Output the specified text (a) in the output.
/NOTITLE	File Control	OUTPUT	Do not add a title to each page.
/NOUNPRE	General	OUTPUT	The /NOUNPRESERVE switch allows any file to be output.
/NULL	General	OUTPUT	Do not output an error message if there are no files in the request and do not create a queue entry. This is assumed at KJOB time.
/OKBINARY	File Control	OUTPUT (LPT)	Print files whose extensions include binary information. Normally files with extensions .SAV, .SHR, .LOW, .REL, and .HGH will not be in print queues.
/OKNONE	File Control	OUTPUT	Do not produce message if no files match the wildcard construction. However, a totally null queue request produces a fatal error message.
/OKPROTECTION	General	ALL	Ignore all protection errors.
/OPTION:S	General	ALL	Use the option line QUEUE:S in the SWITCH.INI file.
/OUTPUT:n	General	INPUT	.LST files and any requests deferred to LOGOUT time (default).
/PAGE:n	General	OUTPUT	Use n (decimal) as the maximum number of pages of output that your job can print. If the switch is omitted, the maximum is 200 pages; if only the value is omitted, a maximum of 2000 pages can be printed.
/PAPER:x	File Control	OUTPUT	Identical to /PUNCH:x., /PRINT:x, /TAPE:x, or /PLOT:x.
/PARITY: EVEN or ODD	File Control	ALL	List the tape with either EVEN or ODD parity.
/PATH.[DIR]	File Control	OUTPUT	Run the job in the default directory.

Switch	Category	Queues	Meaning
/PHYSICAL	File Control	ALL	Suppress logical device names for the specified device.
/PLOT:ASCII	File Control	OUTPUT (PLT)	Plot the file in ASCII mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.
/PLOT:BINARY	File Control	OUTPUT (PLT)	Plot the file in binary mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.
/PLOT:IMAGE	File Control	OUTPUT (PLT)	Plot the file in image mode. If the /PLOT switch is omitted, the file is plotted in the data mode specified in the file.
/PRESERVE			
/PRINT:ARROW	File Control	OUTPUT (LPT)	Convert all control characters to up-arrow format except 011-015 and 020-024. This is the default.
/PRINT:ASCII	File Control	OUTPUT (LPT)	Send the file to the line printer with no changes.
/PRINT:OCTAL	File Control	OUTPUT (LPT)	Print the file in octal.
/PRINT: SUPPRESS	File Control	OUTPUT (LPT)	Convert all carriage-control characters to line-feeds except for ASCII code characters and CR and DC3; this switch implies the use of the /PRINT:ARROW. Equivalent to operator command to spooler (SUPPRESS).
/PRIORITY:n	General	ALL	Give the specified external priority (n = 0 to 62) to the request. A larger number is greater priority. The default is 10 if no switch is given, and 20 if a switch is given without the value.
/PROTECT:nnn	General	ALL	Specify a protection nnn (in octal) for this job or queue entry. If the switch or the value of the switch, is omitted, the standard protection is assumed.

QUEUE Command (Cont.)

Switch	Category	Queues	Meaning
/PUNCH:026	File Control	OUTPUT (CDP)	Punch files in 026 Hollerith code. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:ASCII	File Control	OUTPUT (CDP)	Punch files in ASCII card code. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:BCD	File Control	OUTPUT (CDP)	Same as /PUNCH:026.
/PUNCH: BINARY	File Control	OUTPUT (CDP)	Punch files in check-summed-binary card format. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH:IMAGE	File Control	OUTPUT (CDP)	Punch files in image card format. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/PUNCH: MNEMONIC	File Control	OUTPUT (CDP)	Punch files in mnemonic binary Mode. If the /PUNCH switch is not given, the files are punched according to the data mode of the file.
/REMOVE	File Control	OUTPUT	Remove the file from the queue. This switch is valid only with the /MODIFY switch and can be used to remove a previously submitted file as long as the batch system has not started processing your job.
/REPORT:code	File Control	OUTPUT (LPT)	Prints the specified report cover page and up to one additional page within a COBOL report file. Code can be up to 12 characters in length.
/RESTART: 0 or 1	General	INPUT	A value of 0 (default) means the job cannot be requeued or restarted by the operator after a system crash. A

Switch	Category	Queues	Meaning
			message is sent to the job's log file. A value of 1 means the job will be requeued or restarted. Your job should not be restarted if there are changes to the permanent file directory.
/RUN:file	File Control	OUTPUT	Execute the specified file.
/RUNCORE:n	File Control	OUTPUT	Execute the specified file in nK or core.
/RUNOFFSET:n	File Control	OUTPUT	Execute the specified file with offset n.
/SEQUENCE:n	General	ALL	Specify a sequence number to aid in identifying a request to be modified or deleted.
/SINCE:t	General	OUTPUT	Queue only the files with creation dates after the specified time t where t is in the form dd-mmm-yy hh:mm.
/SPACING: DOUBLE	File Control	OUTPUT (LPT)	Double-space the output lines.
/SPACING: SINGLE	File Control	OUTPUT (LPT)	Single-space the printed lines (default).
/SPACING: TRIPLE	File Control	OUTPUT (LPT)	Triple-space the printed lines.
/STRS	File Control	OUTPUT	Search for the file on all structures in the search list and take each occurrence. The default is to take just the first occurrence of the file.
/TAG:xxx	File Control	INPUT	Start at the statement labeled xxx of the control file. Equivalent to GOTOxxx at the beginning of the control file.
/TAPE:ASCII	File Control	OUTPUT (PTP)	Punch the tape in ASCII code. If the /TAPE switch is not specified, the files are punched according to the data mode of the file.

Switch	Category	Queues	Meaning
/TAPE:BINARY	File Control	OUTPUT (PTP)	Punch the tape in binary mode. If the /TAPE switch is not specified, the files are punched according to the data mode of the file.
/TAPE:IBINARY	File Control	OUTPUT (PTP)	Punch the tape in image-binary mode. If the /TAPE switch is not specified, the files are punched according to the data mode of the file.
/TAPE:IMAGE	File Control	OUTPUT (PTP)	Punch the tape in image mode. If the /TAPE switch is not specified, the files are punched according to the data mode of the file.
/TIME: hh:mm:ss	General	INPUT	Specify the central processor time limit for your job. A colon must be used to separate the hours, minutes, and seconds. If no switch is specified, the limit is 5 minutes; if the switch is specified without a value, the limit is 1 hour.
/TITLE	File Control	OUTPUT	Print a title on each page of output.
/TPLOT:n	General	INPUT	Use n (decimal minutes) as the maximum amount of plotting time allowed for your job. If the switch is omitted, no plotter time is allowed; if the value is omitted but the switch is specified, the maximum plotter time is 10 minutes.
/UNIQUE: 0 or 1	General	INPUT	Run any number of batch jobs under this project-programmer number at the same time, if 0. Runs only one batch job at any one time, if 1 (default).
/UNPRESERVED	General	OUTPUT	Output file only if not preserved.

Characteristics

The QUEUE command (and its associated variations):

Leaves your terminal in monitor mode.
Runs the QUEUE program, thereby destroying your core image.
Does not require LOGIN when only queue listings are desired.

Examples

```
.QUEUE FILEA,FILEB↵
```

Enter file FILEA.LPT and FILEB.LPT in the line-printer queue under the job name of FILEA.

```
.QUEUE INP:=TEST↵
```

Enter file TEST.CTL in the batch input queue under jobname TEST and log file with name TEST.LOG.

```
.QUEUE IN:PAYR=MAN↵
```

Enter file MAN.CTL in the batch input queue under jobname PAYR and log file with name MAN.LOG.

```
.QUEUE DSK:A.X=/LIST↵
```

Place a queue listing of all jobs into file A.X in your disk area.

```
.QUEUE INP:FREED=FILEA/CREATE
  /PRIORITY:4/ TIME: 1:5 ↵
```

Place file FILEA.CTL in the Batch input queue with the job name FREED. An external priority of 4 and CPU time limit of one minute and five seconds are set for your job. The log file is named FILEA.LOG.

```
.QUEUE INP:TEST=/KILL↵
```

Remove the entry corresponding to TEST.CTL from the batch input queue.

```
.QUEUE INP: JOBNAM=/MODIFY/TIME:2:00↵
```

Alter the time parameter of the entry corresponding to JOBNAM.CTL in the batch input queue.

```
.QUEUE INP:=JOB.CTL/PAGES:500/TPLOY:22↵
```

Establish a limit of 500 pages and 20 minutes of plotting on the output generated by this job.

```
.QUEUE PLT:=JOB.PLT/LIMIT:20↵
```

Queue a file to PLTSPL with a limit of 20 minutes of plotting time.

QUOLST Program

Function

The QUOLST program informs you of both the amount of disk space you have used and the amount you have left on each file structure in your search list. This program also returns the amount of free space that the system has left for all users of the structure. Free system space on structures not in your search list is not output. You can obtain this information by typing SYSTAT/F.

The output given for each file structure consists of 1) the structure name; 2) the number of blocks allocated; and 3) the number of blocks left in the logged-in quota, in the logged-out quota, and on the structure. The number of blocks allocated is the same as the number output in the summary of the DIRECT command when the /ALLOC switch is used.

Note that the QUOLST program does not return statistics for a user logged-in under [1,2] because this project-programmer number has infinite quotas.

Command Format

R QUOLST

Characteristics

The R QUOLST command:

Leaves the terminal in monitor mode.
Runs the QUOLST program, destroying your core image.

Examples

```
.R QUOLST↵

USER:   27,4072
STR     USED    LEFT:(IN)       (OUT)   (SYS)
DSKA:   0       1000            100     1388
DSKC:   245     9755            4755    12916
DSKB:   50      9950            4950    92448

.
```

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). This command is the same as RUN SYS:file.ext core and is the usual way to run a system program that does not have a direct monitor command to run it.

This command clears all of user core. However, programs should not count on this action and should explicitly clear those areas of core that are expected to contain zeros (i.e., programs should be self-initializing). This action allows programs to be restarted by a CTRL/C, START sequence without having to do another R command. Note that if a directory specification should be present in the command string, it overrides the assumed SYS: device.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (TWO EOFs in a row).

Command Format

R file.ext core

Arguments are the same as in the RUN command except that SYS; is used as the default device. (In nondisk monitors, the default is the generic name that matches the system device.) (Refer to the RUN command for a discussion of the core argument.)

The extension applies to the low file, not the high file. An extension of .SHR, then .HGH, is assumed for the high file. If the user types an extension of .SHR or .HGH, the extension is treated as a null extension because .SHR and .HGH are confusing as low file extensions.

Characteristics

The R command:

Places the terminal is user mode.
Runs a system program, destroying your core image.

Examples

```
.R PIP↵
*
```

```
.R PIP 5↵
*
```

REASSIGN Command

Function

The REASSIGN command allows one job to pass a device to a second job without having the device go through the monitor device pool (restricted or unrestricted). Both restricted and unrestricted devices can be reassigned. This command, applied to DECtapes, clears the copy of the directory currently in core, forcing the next directory reference to read a new copy from the tape, but does not clear the logical name assignment. If a device is INITed, a RELEASE UUO is performed unless the user issuing the command is reassigning the device to himself.

Command Format

```
REASSIGN dev job
```

dev = the physical or logical name of the device to be reassigned. This argument is required.

job = the number of the job to which the device is to be reassigned. If no job is specified, the device is reassigned to the job issuing the command. This is useful when you want to force the next directory reference to come from the tape instead of core.

A logical name that is also a physical name can be reassigned only if the job issuing the command and the job to which the device is to be reassigned have the same project-programmer number, or the user issuing the command has operator privileges (logged-in under [1,2] or logged-in at OPR). However, a logical name cannot be duplicated; because, two devices cannot have the same logical name.

Characteristics

The REASSIGN coammand:

Leaves the terminal in monitor mode.
Requires core.
Will not operate while a device transmits data.

Restrictions

The job's controlling terminal cannot be reassigned.

Examples

```
.REASSIGN LPT:17↵        Reassign the line printer to job 17.

.REASSIGN CDP:4↵         Reassign the card punch to job 4.
```

REATTA Program

Function

The REATTA program allows you to transfer your job from one terminal to another. Unlike the ATTACH command, REATTA does not require a password or that the terminal be of the same type that LOGIN recognizes in order to run the job. For example, usually a [1,2] job can run only on a local terminal. However, the REATTA program can be used to attach a [1,2] job from a local terminal to a remote terminal.

Before reattaching your job, you should verify that the terminal to which you are attaching is turned on and working properly. Otherwise, it might be difficult to retrieve the job.

Command Format

```
.R REATTA
```

REATTA responds by asking for the new terminal name.

```
TYPE NEW TTY NAME:
```

The user answers with either the new terminal name (e.g., CTY, TTY2) or number (e.g., 2). REATTA then responds with

```
FROM JOB n
```

on the old terminal, and

```
NOW ATTACHED TO JOB n
```

on the new terminal.

Characteristics

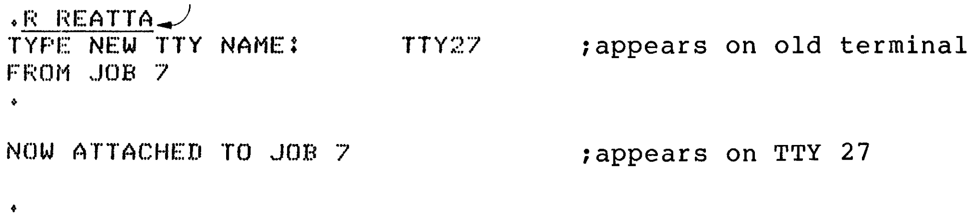
The R REATTA command:

Leaves the terminal in monitor mode.
Runs the REATTA program, thereby destroying your core image.

Restrictions

The R REATTA command is not available to batch users.

Examples



```
.R REATTA↵
TYPE NEW TTY NAME:       TTY27       ;appears on old terminal
FROM JOB 7
.

NOW ATTACHED TO JOB 7                ;appears on TTY 27

.
```

REENTER Command

Function

The REENTER command restarts a program at a location prespecified in the program as an alternate entry point. Although the use of this alternate entry point varies among programs, it is frequently used for a partial reinitialization of the program. (The START command is used for a complete reinitialization and the CONTINUE command is used for no reinitialization.) Note that the DDT command resembles the REENTER command because it also restarts the program at an alternate entry point. However, this point is fixed because it is the beginning address of DDT.

The REENTER command copies the saved program counter value from .JBPC in .JBOPC and starts the program at an alternate entry point specified in .JBREN (must be set by the user or his program). If the job was executing a UUO when it was interrupted (i.e., in exec mode but not in TTY input wait or SLEEP mode), the monitor continues the job until the UUO 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 UUO. If the job is in user mode, the trap also occurs immediately. Therefore, it is always possible to continue the interrupted program after trapping by executing a JRSTF@.JBOPC.

Command Format

REENTER

Characteristics

The REENTER command:

Places the terminal in user mode.
Requires core.
Requires you to have a job number.

Example

```
.REENTER

.
```

RENAME Command[1]

Function

The RENAME command changes one or more items of the file specification of files on disk or DECtape.

Command Format

RENAMEnew1 = old1, new2 = old2,...

If the new file name is specified without an extension, the null extension is assumed. Device or file structure names can be specified only with the new file name and remain in effect until changed or until the end of the command string is reached. In addition, a protection may be specified with the new file name and remains in effect only for that file name. This command accepts the full wildcard construction.

Characteristics

The RENAME command;

Leaves the terminal in monitor mode.
Runs the PIP program, thereby destroying your core image.

Example

```
.RENAME TEST.REL=R
TESTER
FILES RENAME:
? NO FILE NAME DSK: TESTER

.RENAME TESTER=TEST.REL
FILES RENAMED:
TEST.REL

.RENAME WONDER.*=PROG1.*
FILES RENAMED:
PROG1.REL
PROG1.QOR
PROG1.ALG
PROG1.FOR
```

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

RESOURCES Command

Function

The RESOURCES command prints the name of all available devices (except TTYs and PTYs), all file structures, and all physical units not in file structures (unless they are down or nonexistent).

Command Format

RESOURCES

Characteristics

The RESOURCES command:

Leaves the terminal in monitor mode.
Does not require LOGIN.

Example

```
.RESOURCE
DIRS,DSKA,WHTV,FCS,WRS0,DSKN,YELB,DSKC,GAL0,HOUK,DSKB,FSA3,RPA0,RPA1,1,R
PR
PA4,RPA5,RPA6,RPA7,DPA1,DPA4,DPA5,DPA6,RPB4,RPB5,RPB6,RPB7,RPC1,RPC3,RP0
,CDR260,PTR260,DTA262,263,264,265,266,267,MTA260,261,MTB263,MTC260,MTD20

.
```

REWIND Command

Function

The REWIND command rewinds a magnetic tape or a DECtape. This command is equivalent to the PIP command string:

```
dev: (MW)=
```

Command Format

REWIND dev:

dev: = a magnetic tape (MTAn) or a DECtape (DTAn).

Characteristics

The REWIND command:

Leaves the terminal in monitor mode.
Runs the COMPIL program, thereby destroying your core image.

Examples

```
.REWIND DTA014:

.REWIND MTA011:

.
```

RUN Command

Function

The RUN command loads a core image from a retrievable storage device and starts at the location specified within the file (.JBSA).

If the program has two segments, both the low and high segments are set up. If the high file has extension .SHR (as opposed to .HGH), the high segment will be shared. Therefore, if the user RUNs (or GETs) the same program, I/O will not usually be required for the high segment. A two-segment program may have a low file extension (.LOW).

The RUN command clears all your core. However, programs should not count on this action and should explicitly clear those area of core that are expected to contain zeros (i.e., the programs should be self-initializing). This action allows programs to be restarted by a ^C, START sequence without having to do another RUN command.

On magnetic tape, if the low or high segment is missing, a null record is output before the EOF for the missing segment so that two EOFs cannot occur consecutively. Therefore, a saved null segment does not appear as a logical EOT (TWO EOFs in a row).

Command Format

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

dev: = the logical or physical name of the device containing the core image. The default device name is DSK:. (In nondisk monitors, the default is the generic name that matches the system device.)

file.ext = the name of the file containing the core image; .ext applies to the low file, not the high file. An extension of .SHR, then .HGH, is assumed for the high file. If the user types an extension of .SHR or .HGH, the extension is treated as a null extension because .SHR and .HGH are confusing as low file extensions. The default file name is the job's current name as set by the last R, RUN, GET, SAVE, or SSAVE command, the last SETNAM UUO, or the last command which ran a program.

[directory] = the directory name, required only if core image file is located in a disk area other than yours. Note that if a directory specification should be present in the command string, and the device is SYS: it overrides the assumed SYS: device.

core = the amount of core to be assigned to the sum of the low and high segments if different from minimum core needed to load the program or from the core argument of the SAVE command which saved the file.

If core > the minimum low segment size, then an error message occurs.

If core > the minimum low segment size and the sum of the high segment and the minimum low segment size, then the core assignment is the low segment size.

If core > the sum of the minimum low segment and the high segment size, then the core assignment is the size of both the low and high segments to be used.

Core arguments can be specified in units of 1024 words or 512 words (a page) by following the number with K or P, respectively. For example, 2P represents 2 pages or 1024 words. If K or P is not specified, K (1024 words) is assumed.

Note that on KA10 based systems (DECsystem-1040, 1050, 1055), the minimum unit of allocation is 1024 words. Therefore, all arguments are rounded to the nearest multiple of 1024 words (e.g., 3P is treated as 2K on a KA10 based system).

Because previous core is returned, MTA must have the core argument because there is no director telling how much core is for the low segment. (Refer to Appendix D.)

Characteristics

The RUN command:

Places the terminal in user mode.

Restrictions

On systems with a large amount of core memory, you should not specify a core argument that forces the high segment to start higher than 400000 (i.e., a core argument of greater then 128K) unless the program's high segment is location independent. If this is done, the ILLEGAL UUO error message is likely to occur.

Examples

```
.RUN TEST↵

.RUN HITEST<27,7777>↵

.RUN DTA013:TEST1↵

.
```

SAVE Command

Function

The SAVE command function is determined in the process of building the monitor. SAVE can be made identical to either OSAVE or NSAVE.

Example

MONGEN

	M.EXE = 0	M.EXE = 1
SAVE	OSAVE not shared .SAV, .HGH, .LOW, .SHR	NSAVE not shared .EXE
SSAVE	OSSAVE shared .SAV, .HGH, .LOW, .SHR	NSSAVE shared .EXE

Function

The SCHED command types out the schedule bits as set by the operator. You can obtain this information to determine the use of the system (e.g., regular timesharing or batch jobs only) before you LOGIN. The schedule bits are as follows:

```
  0 regular timesharing.
  1 no further logins allowed except from CTY.
  2 no further logins from remote terminals, and no
    answering of data sets.
  4 batch jobs only.
100 device mounts can be done without operator intervention.
200 unspooling allowed.
400 no operator coverage.
```

Command Format

SCHED

Characteristics

The SCHED command:

Leaves the terminal in monitor mode.
Does not require LOGIN.

Example

```
.SCHED↵
000000

.SCHED↵
000400                  Regular timesharing, but no operator
                        coverage.

.SCHED↵
000003                  No LOGINs allowed from local or remote
                        terminals and data sets are not answered.

.SCHED↵
000500                  Regular timesharing, but no operator
                        coverage.  Device mounts can be done without
.                       operator interventions.
```

SEND Command

Function

The SEND command provides a mechanism for one-way interconsole communication. (This command replaces the TALK command.) A line of information is transmitted from one terminal to another, with the identification of the terminal sending the information. With remote communications capabilities, SEND is able to differentiate between stations.

When the SEND command is sent from the central operator's terminal (OPR) or from a terminal logged in as [1,2], it allows a broadcast of a line of information to all non-slaved terminals (including remote terminals) in the system. This allows important information to be dispersed, such as system shutdown or hardware problems. SEND ALL messages do not go to slaved terminals unless the SET TTY NO GAG bit is set to permit reception when the terminal is busy.

A busy test is made on single-destination messages before the message is sent unless the sender or the receiver of the message is OPR or a job logged-in as [1,2]. The receiver of the message is considered busy if their terminal is not at monitor command level. If the receiver is busy, the sender received the message BUSY and the information is not sent, unless the receiving terminal has the TTY NO GAG bit set. (Refer to the SET TTY command.) If the receiving terminal is turned off, the information appears to have been sent, because the hardware cannot detect this condition on hard-wired terminals.

Command Formats

1. SEND dev:text

2. SEND JOB n text

3. SEND m:text

dev = any physical terminal name (CTY included) or OPRnn. If OPRnn is specified, the message is sent to the operator at station nn. If OPR (nn is null) is specified, the message is sent to the operator at your logical station. If the terminal sending the message is the operator's terminal, the argument may be ALL to provide the broadcast operation.

n = the job number to which the message is to be sent.

m = the number of TTY.

the message printed on the receiving terminal appears as follows:

```
;;TTYn: - test
```

where:

n is the TTY sending message, and the text is the message. A bell sounds on the receiving terminal when the message is sent.

Characteristics

The SEND command:

Leaves the terminal in monitor mode.
Does not require LOGIN

Restrictions

The SEND command is not available to the batch user.

Examples

```
.SEND OP: PLEASE WRITE-ENABLE DTA 003:↵
```

SET BLOCKSIZE Command

Function

The SET BLOCKSIZE command sets a default blocksize (in words) for the specified magnetic tape.

Command Format

```
SET BLOCKSIZE dev:nnnn
```

dev: = MTAn:where n is the number of the magnetic tape drive for which the blocksize is to be set, or a logical name associated with a physical magnetic tape. You must have the magnetic tape assigned to you. This argument is required.

nnnn = a decimal number between 3 and 4094 designating the block size for this magnetic tape. No additional checking is done for the legality of the specified number besides the check for the maximum 4094 and a minimum of 3. This argument is required.

Characteristics

The SET BLOCKSIZE command:

Leaves the terminal in monitor mode.

Examples

```
.SET BLOCKSIZE MTA002:3956↵

.ASSIGN MTA004:NAME:↵
MTA004 ASSIGNED

.SET BLOCKSIZE NAME:2000↵

.
```

Function

The SET BREAK command is used (KI10 and KL10 processors only) primarily during the debugging process. It is primarily useful when the program which is being debugged:

1. Will not fail when DDT has been loaded

2. Destroys DDT when DDT is loaded

3. Destroys the contents of a memory location at an unpredictable point during program execution

It is possible to break when the specified location is read from, written into, and/or fetched. It is also possible to break on monitor references to items in the user's address space. This is useful when the monitor is storing or retrieving arguments to/from unexpected locations in your address space because of malformed UUO argument lists.

If you are breaking on a WRITE condition, the write condition causing break will not yet have been executed. Therefore, the instruction located at PC and all operands should be examined before continuing program execution.

Command Format

```
SET BREAK AT adr ON condition

SET BREAK NO condition

SET BREAK NONE

SET BREAK USERS[1]

SET BREAK NO USERS[1]
```

where: AT and ON are optional portions of the command line.

adr is an octal number in the range 0-777777, representing user virtual address.

condition is one or more reason for allowing the break to occur.

Multiple conditions may be specified within one command; these conditions are separated from one another by a comma. The possible conditions which may be specified are:

Condition	Meaning
READ	Break if the contents of adr are read by the program. Note that this condition causes a break to occur on a read-modify write as well as on a read.
WRITE	Break if the location specified by adr is written into.

[1] An operator-privileged command. See Appendix A of the Operator's Guide.

SET BREAK Command (Cont.)

Condition	Meaning
EXECUTE	Break if an instruction is fetched from the location specified by adr.
ALL	Break if the location specified by adr is read from (READ), written into (WRITE), or fetched from (EXECUTE).
MUUO	Break 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 desiring 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.

When issuing a SET BREAK command without specifying any break conditions, the conditions specified in the previous SET BREAK command are still in effect. If there was no previous SET BREAK command, ALL is assumed. When issuing a SET BREAK command without specifying any address, the conditions included in the command line are 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 is effect until removed.

If you wish to remove a break condition, the condition to be removed may be specified in the following command:

```
SET BREAK NO condition
```

If all existing break conditions are to be removed, the following command line may be issued:

```
SET BREAK NONE
```

This command will remove all existent break conditions, but will not remove a previously specified address. An example setting a default break location is:

```
SET BREAK 1000 ON READ, WRITE, EXECUTE
SET BREAK NO READ
SET BREAK NONE
SET BREAK EXECUTE;  1000 IS DEFAULT BREAK LOCATION
```

When a break occurs, one of the messages:

```
%ADDRESS BREAK AT USER PC xxxxxx

%ADDRESS BREAK AT EXEC xxxxxx UUO AT USER xxxxxx
```

will be typed, and the terminal will be left in monitor mode. The second message is produced when MUUO was included in the SET BREAK command line. If the user types:

```
.CONTINUE
```

the program will continue execution at the instruction which caused the break.

Characteristics

Requires LOGIN.

Leaves the terminal in monitor mode.
KI10 and KL10 processors only.

Examples

```
.SET BREAK AT 1000 ON READ,WRITE↵

.SET BREAK EXECUTE↵

.SET BREAK NONE↵

.SET BREAK 1000 MUUO, EXECUTE, WRITE↵
```

SET CDR Command

Function

The SET CDR command sets the file name for the next card-reader spooling intercept. (Refer to DECsystem-10 Monitor Calls.) This command is generally not needed, even when the card reader is being simulated on the disk via the spooling mechanism. It is included in case you wish to reset or change the spooling. In addition, the Batch Controller uses this command to read spooled input card decks.

Command Format

SET CDR filename

filename = one- to three-character file name to be used on next card-reader INIT.

Characteristics

The SET CDR command:

Leaves the terminal in monitor mode.

Examples

```
.SET CDR A↵
.
.SET CDR MAS↵
.
```

SET CPU Command

Function

The SET CPU command allows the privileged user to change the CPUs on which their job can run. It is used in a multiprocessing system to specify whether the programs run under the job can be processed on the primary CPU, the secondary CPU, or either CPU. The job remains with the specified CPU until 1) another SET CPU command with a different specification is given, 2) a KJOB command is issued, or 3) you as privileged user's program override the SET CPU command by issuing the SETUUO with a different specification. If the SETUUO overrides the command, the specification given in the UUO remains in effect until a RESET or EXIT UUO or another SETUUO with a different specification is executed. When an EXIT or RESET UUO is executed, the job reverts back 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 competed for jobs with the ALL specification so that the load is dynamically balanced between CPUs. Therefore, this command is generally not needed but is provided in case the user wishes to change the CPU specification.

Command Formats

1. SET CPU CPxn

   adds the specified CPU to the jobs CPU specification.

2. SET CPU NO CPxn

   removes the specified CPU from the job's CPU specification.

3. SET CPU ALL

   adds all of the CPUs to the job's CPU specification.

4. SET CPU ONLY CPxn

   changes the CPU specification so that it includes only the specified CPU.

x = either U designating a logical name, or A or I designating physical names for a KA10 processor (DECsystem-1055) or KI10 processor (DECsystem-1077), respectively.

n = a decimal number from 0 to the number of processors in the system.

Characteristics

The SET CPU command:

Leaves the terminal in monitor mode.

Restrictions

The privileges required for using this command are determined by bit 5 (JP.CCC) of the privilege word, .GTPRV.

SET CPU Command (Cont.)

Examples

```
    .SET CPU0 ONLY CPU1↵
    .
    .SET CPU CPA0↵
    .
```

SET DEFAULT Command

Function

The SET DEFAULT command sets a default protection code for any files you may create. 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 057, via MONGEN.

Command Format

```
SET DEFAULT PROTECTION ON
SET DEFAULT PROTECTION OFF
SET DEFAULT PROTECTION n
```

The arguments to the SET DEFAULT PROTECTION command are described below.

n is the value the system will use as the protection code for any files you may create.

ON indicates to the system that it is to use your specified protection code (i.e., n).

OFF indicates that the system is to use the system's standard protection code when you create files, even if you have issued a SET DEFAULT PROTECTION n command.

Characteristics

Requires LOGIN.
Leaves your terminal in monitor mode.

Example

```
.SET DEFAULT PROTECTION ON
```

SET DENSITY Command

Function

The SET DENSITY command sets a default density (bits/inch) for the specified magnetic tape.

Command Format

```
SET DENSITY MTAn:density
```

MTAn: where n is the number of the magnetic tape drive for which the density is to be set, or a logical name associated with a physical magnetic tape. You must have the device assigned to you. This argument is required.

```
density =  200 bits/inch
           556 bits/inch
           800 bits/inch
          1600 bits/inch
          6250 bits/inch
```

This argument is required.

Characteristics

The SET DENSITY

Leaves the terminal in monitor mode.

Examples

```
.SET DENSITY MTA005:  556↵
```

SET DSKFUL Command

Function

The SET DSKFUL command controls the treatment of your job when it is attempting output and there is either not space available on the file structure being referenced or your quota for that structure is exceeded.

Command Formats

1. SET DSKFUL ERROR

   Outputting stops and an error condition is passed to the program. Most programs respond to the error condition by issuing an error message and returning the job to monitor level without any opportunity for the user to continue.

2. SET DSKFUL PAUSE

   Output stops and execution of the program is suspended. An error message is printed on your terminal and control of the job is returned to the monitor. Generally, you should employ the SEND command to request assistance from the operator. Execution of the program can be resumed with the CONTINUE command as long as you do not give 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 your accounting file entry specifies PAUSE as your defalt.

Characteristics

The SET DSKFUL command:

Leaves the terminal in monitor mode.

Examples

```
SET DSKFUL:PAUSE↵

.
```

SET DSKPRI Command

Function

The SET DSKPRI command allows the privileged user to set the priority for his job's disk operations (data transfers and head positionings). The standard priority is 0, and the range of permissible values is -3 to +3. This means that a priority lower than the standard can be specified as well as 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. (Refer to DECsystem-10 Monitor Calls.) The priority specified in the SET DSKPRI command remains in effect until 1) another SET DSKPRI command is given with a different priority, 2) a KJOB command is issued, or 3) your program overrides the SET DSKPRI command by issuing a DISK. monitor call with a different priority.

Command Format

SET DSKPRI n

n = a decimal number from -3 to +3 indicating the priority to be associated with the job's disk operations. When n =0, the priority is the normal timesharing priority.

Characteristics

The SET DSKPRI command:

Leaves the terminal in monitor mode.

Restrictions

The privileges required for using this command are determined by bits 1 and 2 of the privilege word, .GTPRV. These two bits specify an octal number from 0-3. The user is always allowed a 0 priority.

Examples

```
.SET DSKPRI 2↵

.
```

SET HPQ Command

Function

The SET HPQ command allows the privileged user to place their job in a high-priority scheduler run queue. With this command, you obtain a faster response and CPU time than in the normal timesharing queues. The job remains in the specified high-priority queue until 1) another SET HPQ command to a different high-priority queue is given, 2) a KJOB command is issued, or 3) your program overrides the SET HPQ command by issuing an HPQ UUO with a different value. If an HPQ UUO overrides the command, the level specified in the UUO remains in effect until a RESET or EXIT UUO or another HPQ UUO with a different value is executed. When an EXIT or RESET UUO is executed, the job is returned to the high-priority queue specified in the SET HPQ command.

Command Format

SET HPQ n

n = a decimal number from 0 to 15 indicating the high-priority queue to be entered. When n = 0, the queue is the normal timesharing run queue. Queue numbers from 1 to 15 are high-priority queues. The number of high-priority queues is an installation parameter and may be less than 15.

Characteristics

The SET HPQ command:

Leaves the terminal in monitor mode.

Restrictions

The privileges required for using this command are determined by bits 6 through 9 of the privilege word, .GTPRV. These four bits specify an octal number from 0-17, which is the highest priority queue you can attain.

Examples

```
.SET HPQ 4↵
.
```

SET PHYSICAL Command

Function

The SET PHYSICAL command is used to specify the maximum current physical page limit (CPPL), if the word LIMIT is included within the command line. (CPPL is described in Chapter 1 of The Monitor Calls Manual.) By including the word GUIDELINE in the command line, the SET PHYSICAL command is used to establish a guideline for the page fault handler. The page fault handler will then use the specified figure as a guideline in determining the exact time a program will go virtual.

Command Format

```
SET PHYSICAL      LIMIT           nP
                  GUIDELNE        nK
```

where: LIMIT and GUIDELINE are alternative portions of the command line. If both are omitted, GUIDELINE is assumed.

1K equals 1024 words, and 1P equals a page of 512 words. If K and P are omitted, K is assumed.

K may be specified within the range 1 to 256K; P may be specified within the range 1 to 512P.

If the command SET PHYSICAL LIMIT is given with a 0 argument, the job will never "go virtual."

Characteristics

Requires LOGIN.

Leaves the terminal in monitor mode.

Example

```
.SET PHYSICAL 100P↵

.SET PHYSICAL GUIDELINE 2K↵

.SET PHYSICAL LIMIT 50K↵

.
```

SET SPOOL Command

Function

The SET SPOOL command adds devices to or deletes devices from the current list of devices being spooled for this job. Spooling is the mechanism by which I/O to or from slow-speed devices is simulated on disk. Devices capable of being spooled are: the line printer, the card punch, the card reader, the paper tape punch, and the plotter.

Command Formats

1. SET SPOOL devl,dev2,...devn

   adds the specified devices to the job's spool list.

2. SET SPOOL ALL

   places all spooling devices into the spool list.

3. SET SPOOL NONE

   clears the entire spool list.

4. SET SPOOL NO devl,dev2,...devn

   removes the specified devices from the job's spool list.

devl,dev2,...devn = physical device names of one or more devices to be added to or deleted from the current spool list. These names are taken from the following list: CDP, CDR, LPT, PLT, PTP.

Characteristics

The SET SPOOL command:

Leaves the terminal in monitor mode.

Restrictions

To unspool devices, the job must have 1) the privilege bit set in .GTPRV, 2) bit 28 (200 octal) set in the STATES word by the operator SET SCHED command, or 3) you must be logged-in under [1,2].

Examples

```
.SET SPOOL CDP:

.SET SPOOL NO LPT:

.SET SPOOL NONE
```

SETSRC Program

Function

The SETSRC program is used to manipulate the job's search list or the system's search list. A search list is defined to be the order of the file structures that are to be searched whenever generic device DSK: is explicitly or implicitly specified by you. This search list is originally defined by the system manager to include the file structures that you can access. With the SETSRC program, you can alter the search list defined for you by adding or deleting file structures.

The search list is of the form

```
fs1/s/s, fs2/s/s, ..., FENCE, ...,fs9/s/s
```

where fs is the name of the file structure and /s is a switch modifying the file structure. The file structures on the left of the FENCE comprise the active search list and represent the generic device DSK for this job. The files to the right of the FENCE comprise the passive search list and represent file structures that were once in the active search list. File structures are kept in the passive search list in order that quotas can be checked on a DISMOUNT or KJOB command. The FENCE represents the boundary between the active and passive search list.

Note that the MOUNT and DISMOUNT commands can also change the job's search list by adding or deleting a file structure. Because the SETSRC program does not create a UFD if one does not exist, the MOUNT command should be used to create a UFD. The name of the new file structure is placed at the end of the search list. (Refer to the SETSRC specification in the DECsystem-10 Software Notebooks for a complete description of the SETSRC program.)

Command Format

R SETSRC

You can then respond with any of the following commands:

Command — Explanation

A

Add one or more file structures to the existing list. The file structures (with any switches) are appended to the beginning or the end or the active search list according to the following specifications:

1. If no asterisk appears in the specifications (e.g., fs1, fs2) or if an asterisk appears before the file structure names (e.g., *,fs1, fs2), the file structures are added to the end of the search list.

Command Explanation

2. If the asterisk follows the file structure names (e.g., fsl, fs2, *,), the file structures are added to the beginning of the search list.

3. If the asterisk appears in the middle of the file structures (e.g., fsl, *, fs2), the file structures before the asterisk are added to the beginning of the search list and the file structures after the asterisk are added to the end.

If the specified file structure is currently in the search list, it is removed and then added in the desired position. Therefore, this command can be used to reorder the search list.

C Create a new search list for this job. Any file structures in the current search list that are not in the new list are moved to the passive search list.

CP Create a new default directory path.

CS Create a new system search list (i.e., the file structure search list for device SYS:). You must be logged in under [1,2] to use this command.

H Obtain information about the available commands.

M Modify the current search list and DSK specification by altering the switch settings for individual file structures. This command does not add or remove file structures from the search list.

R Remove file structures from the search list. They are placed on the right side of the FENCE (passive search list) so that on subsequent LOGOUTs or DISMOUNTs quota limits can be checked. Ersatz devices are not affected by this command.

T Type the search list of the job.

TP Type the default directory path.

TS Type the system search list.

The following switches can be used in the SETSRC command string. Switches that modify file structures must appear immediately after the file structure that they modify. Other switches can appear anywhere in the command string. The switches can be abbreviated as long as the abbreviation is unique.

Switches That Modify File Structures

/CREATE	Allow new files to be created on the file structure.
/NOCREATE	Do not allow new files to be created on the file structure when DSK is specified, but allow files to be superseded. Files can be created on the file structure if the user specifies the file structure name explicitly.
/NOWRITE	Do not allow writing on the file structure for this job (i.e., the file structure is read only).
/WRITE	Allow writing on the file structure.

If no switches are specified, /CREATE and /WRITE are assumed. For compatibility with previous versions of SETSRC, /N is equivalent to /NOCREATE and /R equivalent to /NOWRITE.

Switches That Modify The Directory Path
(used only with the CP command):

These switches can be typed in directly as commands by omitting the CP command and the slash (i.e., /SCAN is equivalent to CP/SCAN).

/NOSCAN	Cancel the scan switch for the directory path.
/SCAN	Set the scan switch for the directory path.

Switches That Modify The DSK Or SYS Specification
(used only with the C and M Commands):

These switches can be typed in directly as commands by omitting the C or M command and the slash (i.e., NOSYS is equivalent to M/NOSYS).

/LIB:[proj,prog]	Set the job's library directory to the UFD [proj,prog] and add it to your DSK specification. This means that if a file is not found in your directories in your search list, the library directory will then be searched for the file.
/NOLIB	Remove the library directory from the user's DSK specification.
/NOSYS	Remove the SYS specification from the user's DSK specification.
/NONEW	Remove the [1,5] directory from the user's SYS specification.
/SYS	Add the SYS specification to the user's DSK specification. This means that if a file cannot be found in the user's

directories in his search list or in his library directory (if /LIB: [proj,prog] has been specified), the system directory [1,4] will then be searched for the file.

/NEW — Add the directory [1,5] to the user's SYS specification. This means that when the system directory is searched, the directory [1,5] will be searched before the directory [1,4].

Characteristics

The R SETSRC command:

Places the terminal in user mode.
Runs the SETSRC program, destroying your core image.

Restrictions

The user must be logged in under [1,2] to create a new system search list. The directory path commands (CP and TP) are meaningful only with the 5.04 monitor and later monitors and only if FTSFD is on.

Examples

```
.R SETSRC↵                        Your search list is defined as
*1↵                               DSKB.

DSKB:,FENCE                       Add DSKA to the end of the search
                                  list.

*A DSKA:↵                         Your search list is defined as
                                  DSKB,DSKA.

*I
DSKB:,DSKA,FENCE                  Add DSKC to the beginning of the
                                  search list.

*A DSKC:,*↵                       Remove DSKA from the search list.

*T↵

DSKC:,DSKB:,DSKA:,FENCE
*R DSKA:↵                         Your search list is defined as
*T↵                               DSKC,DSKB.



*DSKC:,DSKB:,/NOWRITE             Do not allow writing on DSKB.

*M /LIB: [27,5110]↵               Set your library directory to
                                  [27,500] and add it to your DSK
                                  specification.

*SYS↵                             Add SYS: to the user's DSK
                                  specification.

*T↵
/LIB: [27,5110]/SYS DSKC:,DSKB:/NOWRITE,FENCE,DSKA:↵
```

SETSRC Program (Cont.)

Your DSK and SYS specifications are listed first followed by the user's search list.

```
*TS↵
```

```
DSKA:,DSKB:,DSKC:,FENCE
```

The system search list is defined as DSKA,DSKB,DSKC.

SET TIME Command

Function

The SET TIME command sets a central processor time limit for a job. When the time limit is reached, the job is stopped and a message is typed. A timesharing job may be continued by typing CONT, but no time limit is in effect unless it is reset. A batch job cannot be continued.

Command Format

SET TIME n

n = number of seconds of central processor time to which the job is limited. An argument of 0 cancels the time remaining.

Characteristics

The SET TIME command:

Leaves the terminal in monitor mode.

Restrictions

The SET TIME command is ignored in a batch control file. A batch job has its time limit set via the /TIME switch in the QUEUE or SUBMIT command string or on the $JOB card.

Examples

```
.MAKE LOOP.F4↵
```

Create a program with an infinite loop.

```
*I10      CONTINUE↵
          GOTO 10↵
          END↵
<ESC><ESC>
*ESSX<ESC><ESC>
```

```
.TYPE LOOP.F4↵
10        CONTINUE
          GOTO 10
          END
```

Type the program.

```
.LOAD LOOP↵
FORTRAN:  LOOP
MAIN.
LINK*  LOADING

EXIT
```

Compile and load the program.

```
.SET TIME 5↵
```

Set the time limit to 5 seconds.

```
.TIME↵
11.08
11.08
KILO-CORE-SEC-95
```

Clear the incremental run time, so that the SET TIME command can be checked.

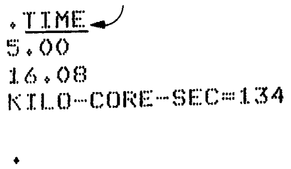
```
.START↵
```

Start the loop.

```
?
?TIME LIMIT EXCEEDED
```

As expected, the time limit was exceeded.

SET TIME Command (Cont.)



```
.TIME
5.00
16.08
KILO-CORE-SEC=134

.
```

Function

The SET TTY command (or TTY command) declares properties of the terminal on which the command is typed to the scanner service. With terminals connected directly to the DECsystem-10, the system manager can set some default conditions, so that this command is usually not needed. However, you are likely to need this command when connected by telephone lines to a dial-up data set, because it may not be possible for the system manager to predict the type of terminal which will be used.

Note that all SET TTY commands can be followed by a TTYn: specification, but these become operator-privileged commands. (See Appendix A of the Operator's Guide.)

Command Formats

1. SET TTY NO word

   equivalent to TTY NO word

2. SET TTY word

   equivalent to TTY word

3. SET TTY TTY:n NO word

   equivalent to TTY TTY:n NO word

4. SET TTY TTY:n word

   equivalent to TTY TTY:n word

NO = the argument that determines whether a bit is to be set or cleared. This argument is optional.

word = the various words representing bits that may be modified by this command. The words are as follows:

SET TTY ALTMODE — Converts the ALTmode codes of 175 and 176 to the ASCII standard escape character 033 (initial state).

SET TTY NO ALTMODE — Restores the individual identity of the codes 175 and 176.

SET TTY BLANKS — Restores multiple carriage return/line feeds and form feeds (initial state).

SET TTY NO BLANKS — Suppresses blank lines (consecutive carriage return/line feeds after the first) and outputs form feeds and vertical tabs as 2 carriage return/line feeds. This is useful for a video display terminal in order to increase the amount of output which fits on the screen.

SET TTY COPY Outputs ^ to controlling terminal. (VT50).

SET TTY or TTY Command (Cont.)

SET TTY NO COPY

Outputs ESC to controlling terminal (VT50).

SET TTY CRLF

Restores the automatic carriage return (initial state).

SET TTY NO CRLF

The carriage return normally output at the end of a line exceeding the carriage width is suppressed.

SET TTY DEBREAK

Notifies the system that the terminal has a feature that allows the computer to lock the keyboard (Model 2741 terminals).

SET TTY NO DEBREAK

Turns off the SET TTY DEBREAK command.

SET TTY ECHO

Restores the normal echoing of each character typed in. Most terminals on the DECsystem-10 are used in full-duplex mode which means that the terminal does not print each character as you type it. Instead the monitor must echo each character typed so that the character will be printed. If a terminal prints ^C when it is typed but does not print any of the non-control characters you type, then this command can be used to establish normal echoing. (Initial setting determined by each installation when constructing its monitor at MONGEN time.)

SET TTY NO ECHO

This command suppresses monitor echoing of input characters. Local copy terminals (i.e., terminals that automatically print each character as the user types it) do not require normal monitor echoing, because the echoing would cause each character typed to be printed twice. This command is used to stop double printing of each character typed on a local copy terminal. (Initial setting determined by each installation when constructing its monitor at MONGEN time.)

SET TTY ELEMENT

Changes the typing element number for Model 2741-type terminals. The elemenc numbers available are:

Element	Meaning
987	APL correspondence
029	Standard correspondence
087	Call 360 BASIC
963	Extended binary
938	BCD
988	APL (EBCD)

Note that the SET TTY ELEMENT n command also sets the terminal as if the SET TTY NO LC command were used.

SET TTY NO ELEMENT

Turns off the SET TTY ELEMENT n command.

SET TTY FILL n

The filler class n is assigned to this terminal. Many different types of terminals are supported by the DECsystem-10 operating system. Some terminals require one or more filler characters to be sent following certain control characters such as line feed (LF) and horizontal tab (HT). The table below illustrates the number of fillers sent for each character and filler class. The filler characters are CR (215 octal for even parity) for carriage return characters (CR, 215 octal) and DEL (RUBOUT, 377 octal) for all other characters. No fillers are supplied for image mode output (initial setting determined by each installation when constructing its monitor at MONGEN time).

SET TTY NO FILL

Equivalent to TTY FILL 0 (initial setting determined by each installation when constructing its monitor at MONGEN TIME). Fillers for output and echoing are determined from the following:

Character	Octal	Number of Fillers for Filler Class			
Name		0	1	2	3
BS	010	0	2	6	6
HT	011	0	1 or 2	0	1 or 2[1]
LF	012	0	2	6	6
VT	013	0	2	6	6
FF	014	0	12	21	21
CR on output	015	0	1	3	3[2]
free CR[3]		0	2	4	4
CRLF on output	015-012	0	3	9	9[4]
XON	021	1	1	1	1
TAPE	022	1	1	1	1
XOFF	023	1	1	1	1
NTAP	024	1	1	1	1

[1] 1 if 0-3 spaces to tab stop; 2 if 4-7 spaces to tab stop.

[2] Output only; no fillers on input.

[3] Refer to the SET TTY CRLF command.

[4] Sum of the fillers output for a CR and LF.

SET TTY or TTY Command (Cont.)

SET TTY FORM

This terminal has hardware FORM (PAGE) and VT (vertical tab) characters (initial setting determined by each installation when constructing its monitor at MONGEN time).

SET TTY NO FORM

The monitor sends eight line feeds for a FORM and four line feeds for a VT (initial setting determined by each installation when constructing its monitor at MONGEN time).

SET TTY GAG

Messages transmitted by the SEND command cannot be received at this terminal unless the terminal is at command level (initial state).

SET TTY NO GAG

Messages transmitted by the SEND command can be received at this terminal even though it is not at command level.

SET TTY HOLD

Outputs ESC[to controlling terminal. (VT50).

SET TTY NO HOLD

Output ESC\ to controlling terminal. (VT50).

SET TTY TTYn:IGNORE

Operator-privileged command. (See Appendix A of the Operator's Guide.)

SET TTY TTYn:NO IGNORE

Operator-privileged command. (See Appendix A of the Operator's Guide.)

SET TTY LC

The translation of lowercase character input to uppercase is suppressed.

SET TTY NO LC

This command causes the monitor to translate lowercase characters to upper case as they are received. Some terminals have only uppercase characters, while others have both uppercase and lowercase. Frequently, it is convenient to have a terminal with both uppercase and lowercase simulate the behavior of one with uppercase only. This command causes the monitor to perform this simulation. The echo sent back by the monitor always matches the case of the characters after translation. By looking at the printout, you can determine what translation was performed by the monitor (initial state).

SET TTY PAGEn

This command gives the user the ability to temporarily suspend system output to a terminal without losing it. The XOFF key (^S) suspends the typeout, and the XON key (^Q) restores it for n lines. The XOFF and XON keys are not echoed and are not sent to the user's program. This command is useful for video display terminals where the user may want to read a page of text before it disappears from the screen. Note that this preempts the use of ^S and ^Q for reading paper tapes. (See SET TTY TAPE.)

SET TTY NO PAGE

The terminal output control ability of the XOFF and XON keys is disabled. The current interpretation of these keys depends on the last SET TTY TAPE command (initial state).

SET TTY RTCOMPATIBILITY

Disables the ^R and ^T features. (See Section 1.3.1 for a description of ^R and ^T.)

SET TTY NO RTCOMPATIBILITY

Turns on the ^R and ^T features. (See Section 1.3.1 for a description of ^R and ^T.)

SET TTY SLAVE

The terminal becomes slaved, i.e., no commands may be typed on the terminal, and the terminal may be ASSIGNed by another user. You can slave your own terminal and you must contact the operator in order to unslave it (initial setting is determined by each installation when constructing its monitor at MONGEN time).

SET TTY TTYn:NO SLAVE

Operator-privileged command. (See Appendix A of the Operator's Guide.)

SET TTY SPEEDn

Set to n baud the speed at which the system sends and receives. (You adjust the speed at the terminal by setting a switch on the terminal.)

SET TTY SPEEDm,n

Set the terminal transmitting speed to n and the receiving speed to m. (This is especially useful for display terminals that permit transmitting and receiving at different speeds.)

SET TTY TAB

This terminal has hardware TAB stops every eight columns (initial setting is determined by each installation when constructing its monitor at MONGEN time).

SET TTY or TTY Command (Cont.)

SET TTY NO TAB

The monitor simulates TAB output from programs by sending the necessary number of SPACE characters.

SET TTY TAPE

The XON key (^Q) causes the terminal to read paper tape. The XOFF key (^S) causes the terminal to stop reading paper tape. (Refer to Chapter 5 of Decsystem-10 Monitor Calls.)

SET TTY NO TAPE

The XON key (^Q) and the XOFF key (^S) have no special paper tape function. They may have a PAGE function (initial state).

SET TTY TIDY

Specifies (for Model 2741 terminals) that every character occupy one print space. The terminal normally types out characters so that they appear the same as you type them in. For example, [prints out as ^<. In TIDY mode, [prints out as <.

SET TTY NO TIDY

Turns off the SET TTY TIDY command.

SET TTY UC

Equivalent to SET TTY NO LC.

SET TTY NO UC

Equivalent to SET TTY LC.

SET TTY WIDTH n

The carriage width (the point at which a free carraiage return is inserted) is set to n. The range of n is 17 (two TAB stops) to 200 decimal. The initial state is 72.

Characteristics

The SET TTY command:

Leaves the terminal in monitor mode.
Does not require LOGIN.

Restrictions

The SET TTY (or TTY) command is not available to the batch user.

SET VIRTUAL LIMIT Command

Function

The SET VIRTUAL LIMIT command is used to specify the current virtual page limit (CVPL). CVPL is described in Chapter 1 of the Monitor Calls Manual.

Command Format

```
SET VIRTUAL LIMIT        nK
                         nP
```

where: LIMIT is an optional portion of the command line.

1K equals 1024 words, and 1P equals a page of 512 words. If K and P are omitted, K is assumed.

K may be specified within the range 1 to 256K; P may be specified within the range 1 to 512P.

If the command SET VIRTUAL LIMIT is given with a 0 argument, the value specified for CVPL is used. CVPL is set by the system administrator and indicates the current virtual page limit.

Characteristics

Requires LOGIN.

Leaves the terminal in monitor mode.
KI10/KL10 processors with virtual memory only.

Example

```
.SET VIRTUAL LIMIT 32K↲

.SET WATCH↲
?ARGS ARE: DAY,RUN,WAIT,READ,WRITE,VERSION,MTA,ALL,NONE

.SET WATCH DAY↲

.SET WATCH RUN↲

.SET WATCH WAIT↲

.SET WATCH VERSION DAY↲

.RUN LOOP↲
<14:26:40>
?LOOP.SAV NOT FOUND
<0.06 0.91  >

.R PIP↲
<14:26*50>

<S:PIP 33B(260) +>
*^C
<0.08 6.08  >

.
```

SET WATCH Command

Function

The SET WATCH command sets the system to print incremental job statistics automatically. This command provides the user with a tool for measuring the performance of his programs.

Command Formats

1. SET WATCH arg(1),arg(2),...,arg(n)

   prints the specified WATCH statistics.

2. SET WATCH ALL

   prints all the WATCH statistics.

3. SET WATCH NONE

   eliminates the printing of all WATCH statistics.

4. SET WATCH NOagr(1),arg(2),...,arg(n)

   eliminates the printing of the specified WATCH statistics.

The following arguments enable printing whenever a monitor comand switches the console from monitor the user mode.

arg = DAY prints the time of day, as [HH:MM.SS]

arg = VERSION prints the version of the program in standard format (refer to the VERSION command).

The following arguments enable printing whenever the console is returned to monitor mode via the ^C, EXIT, HALT, ERROR IN JOBn, or DEVICE xxx OPR zz ACTION REQUESTED messages.

arg = READ prints the incremental number of disk blocks read modulo 4096.

arg = RUN prints the incremental run time.

arg = WAIT prints the wait time (time elapsed since the user started or continued the program).

arg = WRITE prints the incremental number of disk blocks written modulo 4096.

When an UNLOAD is performed, MTA prints magtape statistics in the following form.

```
[MTxn:reelid READ (c/h/s) = a/b/c WRITE (c/h/s) = d/e/f]
```

where: x is an alphabetic representing the tape controller.

n is a number representing the drive number.

reelid is the reel identification.

a is the number of characters read.

b is the number of hard-read errors.

c is the number of soft-read errors.

d is the number of characters written.

e is the number of hard-write errors.

f is the number of soft-write errors.

When a, b, and c are 0, the information pertaining to READ will not be printed.

When d, e, and f are 0, the information pertaining to WRITE will not be printed.

To prevent this message from being printed, the user can type the following command:

```
.SET WATCH NO MTA
```

This message will be produced by an UNLOAD (MTAPE11).

Any combination of the arguments may be specified in any order. Statistics are not printed for commands that do not run programs, such as ASSIGN or PJOB. When you log in your job is set to WATCH the statistics of which you notified the system manager. The information on what statistics to WATCH is kept in ACCT.SYS.

The order of the error messages is the same as the order of output. Therefore, if you forget either the argument or the significance of the statistics you can find these out by typing SET WATCH and examining the message. A single space is always typed between each statistic, whether the statistic appears or not; therefore, it is possible to tell which statistics are being typed.

NOTE

Enabling WATCH output interacts with the incremental data typed by the TIME and DSK commands.

Characteristics

The SET WATCH command:

Leaves the terminal in monitor mode.

Examples

```
.SET WATCH↵
?Args are: day,run,wait,read,write,version,mta,all,none
.SET WATCH DAY↵

.SET WATCH RUN↵

.SET WATCH WAIT↵

.SET WATCH VERSION DAY↵

.R SOS↵
[8:33:46]
```

SET WATCH Command (Cont.)

```
[S:SOS 21(134) + ]

File: ^C
[0.18 10.16   ]

.R PIP↲
[8:34:04]

[S:PIP 33C(344) + ]
*^C
[0.01 6.65   ]

.R TECO↲
[8:34:16]

[S:TECO 24(202) + ]
*^C
[0.03 8.15   ]

.
```

SKIP Command[1]

Function

The SKIP command moves a magnetic tape forward a specified number of files or records or to the logical end of tape. This command, depending on its arguments, is equivalent to the following PIP command strings:

```
MTAn: (M #nA) =
MTAn: (M #nD) =
MTAn: (M #nT) =
```

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

Command Formats

1. SKIP MTAn: x FILES

   advanced forward x files.

2. SKIP MTAn: x RECORDS

   advances forward x records.

3. SKIP MTAn: EOT

   advances forward to the logical end of tape.

The words FILES, RECORDS, and EOT can be abbreviated to F, R, and E, respectively.

Characteristics

The SKIP command:

Leaves the terminal in monitor mode.
Runs the PIP program, thereby destroying your core image.

Examples

```
.SKIP MTA001: EOT↵
.


.SKIP MTA002: 20 RECORDS↵
```

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

START Command

Function

The START command begins execution of a program that you either loaded previously with the GET command or that you interrupted while running (e.g., ^C). The old program counter is copied from .JBPC to .JBOPC. 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 (i.e., it was in exec mode but not in TTY input wait or SLEEP mode), the monitor sets a status bit (UTRP) and continues the job at the location at which it was interrupted before trapping to the specified START address. When the monitor call processing is completed, the monitor clears the status bit, sets .JBOPC to the address following the monitor call, and then traps to the START address. If the job is in TTY input wait or SLEEP mode, the trap to the program occurs immediately, and .JBOPC contains the address of the monitor call. If the job is in user mode, the trap also occurs immediately.

Command Format

START addr

addr is the address at which execution is to begin 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 may specify a starting address of 0.

Characteristics

The START command:

Places your terminal in user mode.
Will not operate while a device transmits data.
Requires core.
Requires LOGIN if you specify an address argument.

Example

```
.START↵
```

Function

The SUBMIT command places entries into the input queue for the batch system. This command is equivalent to the following form of the QUEUE command:

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

Command Format

```
SUBMIT jobname = control file, log file
```

jobname is the name of the job being entered into the queue.

control file is the name of the control file. This file contains all monitor-level and user-level commands for processing by the Batch Controller (BATCON).

log file is the name of the log file. This file is used by the Batch Controller to record its processing of the job.

Only the two files mentioned above can be specified in a request to the batch input queue. The name of the control file is required; the log file name is optional and, if omitted, is taken from the control file. If the job name is omitted, it defaults to the name of the log file, if present, or the name of the control file. If an extension is omitted, the following are assumed:

.CTL for the control file
.LOG for the log file

You can obtain a listing of entries in the input queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., SUBMIT [27,4072]). If no arguments appear in the command string (i.e., only the command name is given), all entries in the batch input queue for all jobs are listed.

Three categories of switches can be used in the command string:

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

2. General - Each switch in this category can appear only once in the command string because these switches affect the entire request. The switch used can appear anywhere in the command string.

3. File control - Any number of these switches can appear in the command string because these switches are specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the filename, it becomes the default for subsequent files. For example, the command string

   ```
   SUBMIT FILEA/DISP:PRE,FILEB
   ```

   indicates that the DISPOSE switch applies only to FILEA. The command string

```
SUBMIT /DISP:PRE FILEA,FILEB
```

indicates that the switch applies to both files.

See the QUEUE command for a complete list of switches and their functions.

Characteristics

The SUBMIT command:

Leaves the terminal in monitor mode.
Runs the QUEUE program.
Does not require LOGIN when only queue listings are desired.

Examples

```
.SUBMIT USRJOB=CONTRL.LOGFIL↵
```

The defaults are as follows:

1. control file name is CONTRL.CTL

2. log file name is LOGFIL.LOG

3. no cards punched (/CARDS:0)

4. maximum core of 25K (/CORE:25)

5. no dependency (/DEPEND:0)

6. control and log files are saved after spooling (/DISPOSE:PRESERVE)

7. no paper tape punched (/FEET:0)

8. all line printer output is spooled with the maximum pages being 200 (/OUTPUT:4,/PAGE:200)

9. priority is 10 (/PRIORITY:10)

10. standard protection is assumed (/PROTECT:nnn (standard))

11. job is not restarted after a crash (/RESTART:0)

12. maximum CPU time is 5 minutes (/TIME:0:05)

13. no plotter time allowed (/PLOT:0)

14. only one job at a time under a given project-programmer number is run (UNIQUE:1)

```
.SUBMIT USRJOB=/MODIFY/FEET:35/CORE↵
```

Modify the original request to include 35 feet as the maximum number of feet of paper tape that the job can punch and 40K of core as the maximum amount of core that the job can use. This command is valid only if the job has not been started yet by the batch system.

```
.SUBMIT USRJOB=/KILL↵
```

Kill the job only if it has not been started by the batch system.

SYSTAT Command

Function

The SYSTAT command runs a system program that prints status information about the system. This information allows you to determine the load on the system before logging-in.

To write the output on the disk as a file with name SYSTAT.TXT, assign device DSK with logical name SYSTAT.

The SYSTAT command types the status of the system: system name, time of day, date, uptime, percent null time (idle plus lost time), number of jobs in use.

It types the status of each job logged-in: job number; project-programmer number (**,** = detached, [OPR] = the project-programmer number of the operator, [SELF] = user's project-programmer number): terminal line number (CTY = console terminal, DET = detached, Pn = PTY number); program name being run; program size; job and swapped state (refer to DECsystem-10 Monitor Calls); run time since logged-in.

It types the status of high segments being used: name (PRIV = nonsharable, OBS = superseded); device or file structure name from which the segment came; directory name (**,** if detached); size (SW = swapped out, SWF = swapped out and fragmented, F = in core and fragmented on disk, SPY = user is executing the SPY monitor call); number of user in core or on the disk.

Included in the command's response is the amount of swapping space used, the virtual core used, swapping ratio, active swapping ratio, virtual core saved by sharing, average job size.

It types status of busy devices: device name, job number, how device is assigned (AS = ASSIGN command, INIT = INIT or OPEN monitor call, AS+INIT = both ways).

It types system file structures: free blocks, mount count, single-access job.

It types remote stations: number of station, status of station.

It types dataset control: number of the TTY, status of TTY.

Command Format

SYSTAT arg

arg is one or more single letters (in any order) used to type any subset of the SYSTAT output. This argument is optional and if you omit it, the entire SYSTAT output is listed. The following lists the various arguments to the SYSTAT command.

B	busy device status
C	continuous SYSTAT
D	dormant segment status
E	non-disk error report
F	file structure status
G	other system status
H	help text listing the arguments
J	job status
L	list the SYSTAT output on LPT

N	non-job status (i.e., all information except J)
P	disk performance
R	remote station status
S	short job status
T	dataset status
V	if using a display (for paged output)
X	read the file DSK:CRASH.EXE if found, otherwise read specified crashed monitor written in EXE format.

Meanings of job state codes:

RN	in a run queue
WS	I/O wait satisfied
TS	TTY I/O wait satisfied
DS	Disk I/O wait satisfied
AU	Disk alter UFD wait
PS	Paging I/O wait satisfied
MQ	disk monitor buffer wait
DA	disk storage allocation (sat block) wait
CB	disk core block scan wait
D1	DECtape control wait
D2	2nd DECtape control wait
DC	data control wait
CA	core allocation wait (to be locked)
EV	exec virtual memory wait
IO	I/O wait
TI	TTY I/O wait (input)
DI	disk I/O wait
SL	sleep wait
NU	null state
ST	stop (^C) state
JD	daemon wait
^D	daemon wait
TO	TTY output
^C	job stopped
^W	command wait
OW	operator wait
HB	hibernate state
PI	paging I/O wait
EW	event wait
NA	nap (short sleep)

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 indicated job (i.e., job n). A period causes information for your job to be output.

A project-programmer number specified in square brackets causes information to be output only for jobs with the specified project-programmer number. The project and/or programmer number can be wild (i.e., specified with an asterisk).

A number preceded by a number sign (#n) causes information to be output only for jobs from the indicated terminal (i.e., TTYn). In addition, a C following the command indicates CTY, Pnn indicates PTYnn, Tnn indicates TTYnn, and a period indicates the terminal on which the SYSTAT command is issued.

```
Characteristics
     The SYSTAT command:

          Leaves your terminal in monitor mode.
          Runs the SYSTAT program, destroying your core image.
          Does not require LOGIN.

Examples
```

```
.SYSTAT↲

Status of      SCLIZA KL10 SYS#1026 at 11:25:52 on 04-Aug-77

Uptime 1:48:39+1:43:47, 10+61%Null time = 10+38%Idle + 0+23%Lost
66 Jobs in use out of 160.  64 logged in, 3 detached out of 160.  (LOGMA
X)

Job     Who      Line#     What    Size(P)   State    Run Time

 1     [OPR]     P0J5     LPTSPL   26+12    SL              48
 2     [OPR]      1       SYSTAT   13+SPY   ^C              53
 3     37,4620   46       SOS      11+16    TI              11
 4     37,5116   50       SOS      11+16    TI              32
 5     [OPR]      4       OPSER    2+5      HB               5
 6     [OPR]     DET      DAEMON   24+SPY   SL               9
 7     10,2100   11       PPN      1        ^C              18
 8     30,4676   125      TECO     5+8      TI            2:51
 9     [OPR]     CTY      OPSER    2+5      HB  SW           1
10     [OPR]     107      SYSTAT   19+SPY   ^C  SW          41
11     [OPR]     P2J5     LPTSPL   26+12    SL              56
12     31,4775   27       TECO     5+8      TI               8
13     50,3615   34       TECO     5+8      ^C              57
14     [OPR]     P3J5     OPROMO   6+13     SL               5
15     [OPR]     DET      FILDAE   12       HB               6
16     [OPR]     P5J5     BATCON   21+12    HB              10
17     [OPR]     P10J9    DTELDR   7        HB               0
18     [OPR]     DET      MIC      4+14     SL               6
19     30,4661   P12J46   SETSRC   3        ^C  SW           0
20     [OPR]      0       BACKUP   26       EW            1:36
21     [OPR]     P1J9     SYSINF   10       HB               1
22     [OPR]     P4J9     QUASAR   28+12    HB               7
23     [OPR]     P6J9     JUSTIF   29+30    SL               3
24     37,4537   51       DIRECT   6+21     ^C               3
25     [OPR]     P7J9     ALCOMP   12+15    SL               2
26     37,2770   52       QUEUE    14+18    ^C              13
27     12,13     264      TECO     5+8      ^C              57
28     37,4562   227      SOS      11+16    DA            2:54
29     12,2504   263      QUOLST   1        ^C              24
30     [OPR]      6       BACKUP   24       RN            1:03
31     10,2162   116               3+8      ^C              27
32    226,4563   P14J64   QUASAR   12       HB               0
33     30,4661   P11J46   QUASAR   25       HB  SW           0
34    226,4646   131      VED      4+13     TI               1
35    226,4563   P15J64   LOGCSP   1+3      ^C               2 #
36     10,5064   253      FILCOM   4+4      ^C            5:24
37     35,5135   325      OPSER    2+5      ^C              32
38     60,3733   134      IQSORT   46+44    ^C               2
39     30,5134      DDT      20       TI             3
40    226,4563   P16J64   MCS-10   22+40    NA               1
41     16,107    266      QUOLST   1        ^C  SW           2
42     30,3727   126      NEWMAC   23+22    TI            3:03
43   3071,1741   301      QUEUE    14+18    ^C  SW           2
44     10,176    245      QUEUE    14+18    ^C            1:13
45     30,4661   P13J46   LPTSPL   21+7     TI  SW          10 #
46     30,4661   335      OPSER    2+5      HB               2
```

```
47    226,4563    P17J64  LOGCSP   1+3      ^C             0 #
48    226,5071    37      TESTD3   21+44    TI             2
49    226,4563    P21J64  LOGCSP   1+3      ^C             0 #
50     31,1771    115     TECO     7+8      ^C            13
51     27,5031    53      SOS      11+16    ^C            51
52    226,4563    P20J64  LOGCSP   1+3      ^C             0 #
53    400,2040    307     DIRECT   6+21     ^C            10
54     50,2365    235     SYSTAT   19+SPY   ^C  SW         8 $
55     30,3024    265     SIRUS    31+34    TI             1
56     10,466370  304     DIRECT   6+21     TO             4
57     [SELF]     41      SYSTAT   16+SPY   RN             8 $
58     12,5065    316     SOS      11+16    ^C             4
59     31,4577    110     FORTRA   17+51    TI            36
60     50,4035     2      COMPIL   5+8      MQ            33
61   3071,1741    225     BACKUP   12+14    TI          3:00
62    226,4563    P22J64  LOGCSP   1+3      ^C             0 #
63     30,4566    223     SIRUS    31+34    ^C  SW         2
64    226,4563    130     XSER     11+9     SL             4
65      2,5       255     SYSTAT   19+SPY   TO             0 $
66    226,4563    P23J40  LOGIN    3+21     RN             0 $
# means non-system Hi-Seg
$ means Execute Only
Jnn is the controlling Job, Pnn corresponds to TTY423+nn

.
.
.

Busy devices:
Device   Job      Why      Logical

TTY36    48       as+init
DET      48       as
TTY133   40       init
MPX3     40       init
MPX2     40       init
MPX1     40       init
LPT260   11       as+init
LPT261    1       as+init
DTA261    7       as
MTB260   30       as+init
MTB263   61       as+init
MTB264   20       as+init
'LL      60       as
'LL      13       as
'LL      39       as
123 disk DDBs

System File Structures:
Name     Free     Mount
GALO     82462    2
DSKA     2032     18
TATT     1595     0
FTN      26530    1
TAUX     2800     1
RLD      51297    2
DSKN     18975    26
YELB     68360    0
DSKC     277670   65
CCBL     36842    2
DSKB     243095   69
Total Free 811658


.
```

TECO Command[1]

Function

The TECO command opens an already existing file on disk for editing with TECO. Refer to the TECO manual in the DECsystem-10 Software Notebooks.

Command Format

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

dev: 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 EDIT-class command are used.

[directory] is the directory name in which the file appears. If omitted, your 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 on to TECO.

Characteristics

The TECO command:

Places your terminal in user mode.
Runs the TECO program, destroying your core image.

Example

```
.TECO LOGOUT.MIC↵

[2K Core]
*
```

```
.TECO PRINT.MIC↵

[2K Core]
*
```

[1] This command runs the COMPIL program, which interprets the commands before running TECO.

TIME Command

Function

The TIME command causes the system to type the total running time since the last time you issued the TIME command, followed by the total running time used by the job since it was initialized (logged-in), followed by the integrated product of running time and core size (KILO-CORE-SEC=). Time is typed in the following format:

```
hh:mm:ss.hh
```

where

```
hh = hours
mm = minutes
ss.hh = seconds to nearest hundredth.
```

Interrupt level and job scheduling times are charged to whichever user was running the system job when the interrupt or rescheduling occurred.

NOTE

> If automatic runtime is enabled via the SET WATCH command, the incremental runtime is usually 0.

Command Format

TIME job

job is the job number of the job whose timing is desired. If job is omitted, the job to which the terminal is attached is assumed. In this case, the monitor types out the incremental runtime (runtime since the last TIME command) as well as the total runtime since the job was initialized.

Characteristics

The TIME command:

Leaves your terminal in monitor mode.
Does not require LOGIN when requesting time for another user's logged in job.

Example

```
.TIME↵
4.55
21.36
KILO-CORE-SEC=175

.TIM↵
0.00
21.36
KILO-CORE-SEC=175
```

The command is given for the first time after LOGIN; therefore, the incremental time equals the total time since LOGIN.

```
.DIR/F↵

WONDER.REL        DSKC:    <27,4072>
WONDER.QOR

.
.
.


.TIME↵
0.70
22.06
KILO-CORE-SEC=183
```

TPUNCH Command

Function

The TPUNCH command is used to place entries into the paper-tape punch output queue. This command is equal to the following form of the QUEUE command:

```
QUEUE PTP:  jobname = list of input specifications.
```

Command Format

```
TPUNCH jobname = list of input specifications
```

jobname is the name of the job being entered into the queue. The default is the name of the first file in the request not the name of the first file given. These differ when the first file given does not yet exist.

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

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

file.ext is the name(s) of the file(s). The file name is optional. The default for the first filename is *, the default for subsequent files is the last filename used. The extension can be omitted; the default is .PTP.

[directory] is a directory to which you have access; your directory is assumed if you do not specify a directory.

The wildcard construction can be used for the input specifications.

You can obtain the listing of entries in the paper-tape punch queue for specific project-programmer numbers by following the command with the desired project-programmer numbers enclosed in square brackets (e.g., TPUNCH [400,27]). If no arguments appear in the command string (i.e., you give only the command name), all entries in the paper-tape punch queue for all jobs are listed.

You can specify switches as part of the input specification that aid in the construction of the queue entry. These switches are divided into three categories:

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

2. General - Each switch in this category can appear only once in the command string because these switches affect the entire request. The switch used can appear anywhere in the command string.

3. File control - Any number of these switches can appear in the command string because they are specific to individual files within the request. The switch used must be adjacent to the file to which it applies. If the switch precedes the

filename, it becomes the default for subsequent files. For example, the command string

```
TPUNCH FILEA, FILEB/DISP:REN, FILEC
```

indicates that the DISPOSE switch applies only to FILEB. The command string

```
TPUNCH /DISP:REN FILEA, FILEB, FILEC
```

indicates that the DISPOSE switch applies to all three files.

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

See the QUEUE command for a complete list of switches and their functions.

Characteristics

The TPUNCH command:

Leaves the terminal in monitor mode.
Runs the QUEUE program, thereby destroying your core image.
Does not require LOGIN when only queued listings are desired.

Examples

```
.TPUNCH SENDMP.REL/TAPE:BINARY/COPIES:3↵
```

Punch 3 copies, in binary mode, of the file DSK:SENDMP.REL.

TYPE Command[1]

Function

The TYPE command directs PIP to type the contents of the named source file(s) on your terminal. Note that if more than one file is requested in the command string, the files are typed out one after another with no indication of the beginning and ending of a file.

Command Format

TYPE list

list is a single file specification or a string of file specifications separated by commas. The file name (including any extension) is required for a directory service.

In addition, the full wildcard construction can be used for the file name and/or the extension.

Switches can be passed to PIP by enclosing them in parentheses in the TYPE command string. When COMPIL interprets the command string, it passes the switches on to PIP.

Characteristics

The TYPE command:

Leaves the terminal in monitor mode.
Runs the PIP program, destroying your core area.

Examples

```
.TYPE LOGOUT.MIC↵
delete *.qno
delete *.mem
dir

.TYPE PRINT.MIC↵
PRINT LL:=COM1.MEM
PRINT LL:=COM11.MEM
PRINT LL:=COM12.MEM
PRINT LL:=COM2A.MEM
PRINT LL:=COM2B.MEM
PRINT LL:=COM2C.MEM
PRINT LL:=COM3.MEM

.TYPE SWITCH.INI↵
00100   sos/save:10/isave:40

.
```

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

UNLOAD Command

Function

The UNLOAD command rewinds and unloads a tape, either magnetic tape or DECtape. When unloading magnetic tape, WATCH statistics are printed on the operator's terminal. These statistics are also printed on your terminal by default.

Command Format

```
UNLOAD dev:

     dev:  is a magnetic tape (MTAn) or a DECtape (DTAn).
```

Characteristics

The UNLOAD command:

Leaves the terminal in monitor mode.
Runs the COMPIL program, destroying your core image.

Examples

```
.UNLOAD DTA017↵
.


.UNLOAD MTA013↵
.
```

USESTAT Command

Function

The USESTAT command returns status information pertinent to your job. This status information is printed on one line in eight columns. The information given includes:

incremental day time in seconds
incremental run time in seconds
incremental disk reads
incremental disk writes
program name
core size
job state
program counter (PC)

Refer to Section 1.3.1 for a description of the status information returned as a result of the USESTAT command (or ^T). An ampersand (&) is printed out after the JOBSTATE if the job is loaded in core.

The results obtained from issuing the USESTAT command may also be obtained by issuing a CTRL/T (^T). CTRL/T can be issued at user level as well as at monitor level.

The PAGE function .PAGEA returns more information (see UUOSYM).

Command Format

USESTAT (or CTRL/T)

Characteristics

CTRL/T

Does not change the state of the terminal.

USESTAT:

Leaves the terminal in monitor mode.

Example

```
.US↵
DAY: :37:57 RUN: 4.85 RD:1111 WR:775 SOSX 8+23P  ^C PC:404570

.
```

VERSION Command[1]

Function

The VERSION command prints the version number of the program in your core area (i.e., the last program you ran implicitly or explicitly). One use of this command is to determine the program that output a message to you. If the terminal is still in user mode (i.e., a character other than a period was output) after the message, you can type the following

```
^C        (two ^Cs if the program is not waiting for input)
.VERSION
```

The monitor returns with the name of the program in core (i.e., the one presumed to have output the message) and the version number of that program. After receiving the information, you can type .CONT to return the terminal to user mode. If the message was a fatal message (i.e., a period was output after the message). CTRL/C need not be typed since the terminal is already in monitor mode. In most cases, you cannot type .CONT after a fatal error message.

The version number is obtained from .JBVER and .JBHVR in the job data area and is printed in standard format. Similar output is automatically generated by the SET WATCH VERSION command (refer to the SET WATCH command description). The output from these two commands is in one of the following representations:

low + high — The low and high segments are different.

low — There is only a low segment.

low + — The low and high segments are the same.

+[1] — A GETSEG monitor call has been performed to a high segment that matches the low segment.

+ high[1] — A GETSEG UUO has been done to a high segment that does not match the low segment.

blank[1] — The high segment has been released.

With the VERSION command, the low and high segments are represented in the format

```
name version
```

With the SET WATCH VERSION command, the low and high segments are represented in one of three formats:

name version — The program is not from SYS;

:name version — The output is the result of a SETNAM UUO (e.g., at the end of loading).

S:name version — The program is a program loaded from the system device (actual SYS: not logical device SYS:)

[1] Output only from the SET WATCH VERSION command.

The name is a SIXBIT name and the version is in standard format. When printing the version number, the standard format is:

major version minor version (edit) - group who modified program last

The major version is octal; the minor version is alphabetic; the edit is octal and enclosed in parentheses; and the group who last modified the program is octal and preceded by a hyphen (0 = DEC development, 1 = all other DEC personnel, and 2-7 = customer use). There are no spaces separating the items, and if an item is zero, it does not appear in print. The parentheses and hyphen also do not appear in print if the corresponding item is zero. The following are examples of version numbers output in standard format.

10B(335)-1	major version 10, minor version B, edit number 335, group that modified program last 1.
7(5)	major version 7, minor version 0, edit number 5, group that modified program last 0.
54A	major version 54, minor version A, edit number 0, group that modified program last 0.

When running under a virtual memory system, DAEMON will be called on execution of the VERSION command if the first page of the high segment is paged out.

Command Format
VERSION

Characteristics
The VERSION command:

Leaves the terminal in monitor mode.

Examples

```
.R TECO↵

*^C

.VERSION↵
TECO 24(202) +

.R SYSTAT↵

^C
.VERSION↵
SYSTAT 473A(223)

.R SOS↵

^C
.VERSION↵
SOS 21(134)

.INITIA↵

^C

.VERSION↵
INITIA 7(146)
```

WHERE Command

Function

The WHERE command enables you to determine:

1. the node-id
2. the node number
3. the software I.D. (monitor name)
4. the creation date of the monitor software

of the node at which a specific peripheral device is located. If the node of a particular terminal is requested, the information returned is that of the physical location of the terminal. The information may or may not be that of the controlling job. This depends on whether or not you have changed your job's logical location by means of the LOCATE command.

When the WHERE command does not provide sufficient user information, the following additional inputs are required:

1. physical device name (both generic and logical);
2. physical line number (for network TTYs);
3. NOT CONNECTED entry (if the network TTY is not connected.)

Command Format

WHERE devn:

dev is any physical name and n is the unit number.
where OPR: allows you to find the location of the controlling job thru the OPSER program functions.

Characteristics

The WHERE command:

Leaves the terminal in monitor mode.
Does not require LOGIN.
Allows for an optional colon after the device name.

Examples

```
.WHERE LPT260

NODE    KL1026(26)      SCLIZA KL10 SYS#1026 07-24-77 LPT260
.
```

ZERO Command[1]

Function

The ZERO command clears the directory of the output device. This command is equal to the following PIP command string:

```
dev:  /Z=
```

Command Format

ZERO dev:

dev: is a DECtape (DTAn) or a disk (DSK). This argument is required.

A directory name can be specified with ZERO DISK: and if you have access to the specified directory, the directory is zeroed. If you do not specify a directory, your directory is assumed.

The only acceptable abbreviation of this command is ZER.

Characteristics

The ZERO command:

Leaves the terminal in monitor mode.
Runs the PIP program, destroying your core image.

Examples

```
.ZERO DTA014:

.ZERO DSK:

.ZERO DSK:<27,4072>

.




.
```

[1] This command runs the COMPIL program, which interpret the command before running the PIP program.

CHAPTER 3

SYSTEM DIAGNOSTIC MESSAGES AND ERROR CODES

The following conventions are used in describing the system diagnostic messages:

dev	a legal device name.
file structure name	a legal file structure name.
file.ext	a legal filename and extension.
adr	a user address.
n	a number.
abc	a disk unit or drive.
x	an alphabetic character.
switch	a switch.

Most messages returned to you fall in one of three categories. These categories are determined by the beginning character of the message.

? at the start of the message indicates a fatal error message.
% at the start of the message represents an advisory or warning message.
[at the beginning of the message indicates a comment line.

(Note that the ONCE-only messages have been removed from this section and placed in ONCE.RNO in the DECsystem-10 Software Notebooks.) The user can also employ the VERSION monitor command to determine the program that output the message. If the terminal is still in user mode (i.e., a character other than a period was output after the message), type the following

```
^C        (two ^Cs if the program is not waiting for input)
.VERSION
```

The monitor returns with the name of the program in core (i.e., the one presumed to have output the message) and the version of that program. After receiving the information type .CONT to return the terminal to user mode. If the message was a fatal message (i.e., a period was output after the message), ^C need not be typed since the terminal is already in monitor mode. In most cases, the user cannot type .CONT after a fatal error message.

The descriptive text given with the message indicates what action you should take when you receive the message. You can, if necessary, notify the operator of any problems by issuing the SEND, PLEASE, or R GRIPE command.

3.1 SYSTEM DIAGNOSTIC MESSAGES

The typein is typed back preceded and followed by ?.

The monitor encountered an incorrect character (e.g., a letter in a numeric argument). The incorrect character appears immediately before the second ?.

For example:

```
.CORE ABC
?CORE A?
```

ACCOUNTING SYSTEM FAILURE...

A program could not append an entry to the accounting file. Notify the operator. (LOGIN, LOGOUT)

?ADDRESS CHECK FOR DEVICE dev

1. The monitor checked a user address on a UUO and found it to be too large (>C(.JBREL)) or too small (<C(.JBPFI)); in other words, the address lies outside the bounds of the user program.
2. The SAVed file is too large for the core assigned, or the file is not a core image file. (GET)

?ALREADY ASSIGNED TO JOB n

The device is already assigned to another user's job (job n).

unit-id ALREADY MOUNTED ON unit

You requested a drive of an already mounted file structure to be mounted or you requested a drive other than the one on which the file structure is mounted. (MOUNT)

?AMBIGUOUS ABBREVIATION

A command or switch has been abbreviated to the point that it is not unique. (COMPIL)

?dev: APPARENT UFD PROTECTION FAILURE

Your attempt to delete a file has failed because of a UFD protection. As a result, KJOB exits to monitor level. (KJOB)

?ARGS ARE: DAY, RUN, WAIT, READ, WRITE, VERSION, ALL, NONE

You either did not type an argument or typed an illegal argument in the SET WATCH command string.

dev:ASSIGNED

The device has been successfully assigned to the user's job.

?ASSIGNED TO JOB n(1), n(2), ...

If there is more than one device of the type specified, the numbers of the other jobs that have the same type of device are output, unless the user assigning the device has all the devices of the specified type. In this case, ?DEVICE ASSIGNED TO JOB is output.

?ATTACH TO USER JOB FAILED

DAEMON could not attach to the user's job. (DAEMON)

?BAD DIRECTORY FOR DEVICE DTAn

The system cannot read or write the DECtape directory without getting some kind of error. This error often occurs when the user tries to write on a write-locked tape or use a DECtape that has never been written on.

?BATCH ONLY

The command issued can only be given by a batch job.

BLOCK NOT FREE

M specifies a unit or file structure logical block that is not free. (ALCFIL)

n BLOCKS ALREADY ALLOCATED

The file already exists. The new specification replaces, rather than updates, the old specification. (ALCFIL)

?n1K BLOCKS OF CORE NEEDED

Your current core allocation is less than the contents of .JBFF.

?BOMB OUT

The location within INITIA that detected the error will be in AC 15 and the console lights. (INITIA)

?BOOTSTRAP LOADER IS NOT IN COPY; TRY/L

An attempt was made to write the bootstrap loader onto a DECtape via the /T switch before the loader was loaded into a core buffer and preserved with the COPY core image. (COPY program)

?BOOTSTRAP LOADER WILL NOT FIT IN 3 BLOCKS

Your bootstrap loader is too big to fit into blocks 0, 1, and 2 of the output DECtape. (COPY program)

?BUFFER CAPACITY EXCEEDED AND NO CORE AVAILABLE

The buffer is not large enough to handle the number of lines required for looking ahead for matches, and additional core is not available. (FILCOM)

?BUSY

The terminal addressed is not communicating with the monitor (i.e., it is accepting a command or returning output from a command). The operator's terminal is never busy. (SEND, JCONT)

?CANNOT DO I/O AS REQUESTED

Input (or output) cannot be performed on one of the devices specified for input (output). For example, input may have been requested for a device that can only do output. (MAKLIB)

?CANNOT DO OUTPUT TO DEVICE dev

Output was attempted to a device that can only do input, or to a device assigned a logical name. (QUEUE)

%CANNOT OVERRIDE PPN IMPLIED BY DEVICE

You specified a project-programmer number in a file specification in his FILE F or FILE Z command that is in conflict with the project-programmer number associated with the device. For example, the file specification, NEW:POOH.BAH[12,144], would generate this message since "NEW:" implies the project-programmer number [1,5]. The project-programmer number specified by you is ignored. (FILE)

?CANNOT PROCESS EXTERNAL SYMBOLS

External symbols were encountered while loading the bootstrap loader with the /L switch. (DTCOPY program)

?CANNOT PROCESS HIGH SEG'S

While loading the bootstrap loader with the /L switch, high segment code was encountered. (DTCOPY program)

?CANNOT REATTACH FROM A BATCH SUBJOB

Batch jobs are not allowed to reattach their jobs. (REATTA)

?CAN'T ACCESS QUEUE - name MUST NOT BE A LOGICAL NAME

The structure named contains the operator request queue (3,3.UFD) and must not be the logical name for some other structure. (MOUNT)

?CAN'T ACCESS QUEUE - file structure name MUST NOT BE LOCKED OR WRITE PROT.

The named structure is being used to queue requests to the operator and therefore may not be locked or write-protected. SETSRC may be used to change the protection. (MOUNT, DISMOUNT, FILE)

?CAN'T ACCESS SYSTEM FILES

ACCT.SYS could not be read. Only the operator may LOGIN until ACCT.SYS is ready. Consult the operator. (LOGIN)

?CAN'T ADD TO YOUR FILE STRUCTURE SEARCH LIST n

n is the error code from STRUUO when trying to add a file structure to search list. (LOGIN)

?file-structure name CAN'T ADD TO YOUR FILE STRUCTURE SEARCH LIST n

"n" is the error code returned from the STRUUO which is trying to add this structure to the user's search list. See the STRUUO UUO error codes in the DECsystem-10 Monitor Calls manual and in the UUOPRV specification. This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. Consult the system administrator. (MOUNT)

?CAN'T ATT TO JOB

The project-programmer number specified is not that of the owner of the desired job, the project-programmer number was not given when it was required, or the PASSWORD given was incorrect. (ATTACH)

?dev CAN'T BE REASSIGNED

1. The job's controlling terminal cannot be reassigned, or
2. the logical name would be duplicated, or
3. the logical name is a physical device name in the system and the job reassigning the device is either logged-in under a different project-programmer number or does not have operator privileges. (REASSIGN)

?CAN'T CONTINUE

The job was terminated due to

1. all ERROR IN JOB messages (except for HALT),
2. the EXIT UUO,
3. the CLOSE command, or
4. the REA command when the device was INITed, and the user attempted to continue his program at the point at which I/O was terminated. The job cannot be continued.

CAN'T CREATE NEW FILE STRUCTURE SEARCH LIST

The monitor cannot create a new file structure search list.

?CAN'T DECIPHER THAT

There is a syntax error in the command string. (MOUNT, DISMOUNT, FILE)

CAN'T DET DEV

The user is not logged-in under [1,2].

?CAN'T ENTER OUTPUT FILE n file descriptor

The ENTER to write the output file failed; n is the disk error code. (DUMP)

?CAN'T EXPAND TABLE xxxx

The DUMP program ran out of core in attempting to expand the indicated table. (DUMP)

?CAN'T FIND INPUT FILE n file descriptor

DUMP cannot locate the file specified as the input file; n is the disk error code. (DUMP)

?CAN'T FIND FILE file.ext

The specified file could not be found.

?CAN'T FIND STRLST.SYS

The system is unable to locate or access STRLST.SYS, the system file that contains necessary information about the structures known to the system. Consult the system administrator. (MOUNT)

?CAN'T GET SWAPPING PARAMETERS

DAEMON tried to obtain the job's swapping parameters and failed. (DAEMON)

?CAN'T GET SWAPPING POINTER FOR JOB

DAEMON tried to obtain the pointer to the user's job on the swapping space and could not because the GETTAB UUO failed. (DAEMON)

?CAN'T GET USERS PPN

DAEMON tried to obtain the user's project-programmer number and could not because a GETTAB UUO failed. (DAEMON)

?CAN'T OPEN file structure name

The file structure is mounted but cannot be opened. No UFD is created, though one may already exist. (LOGIN)

?file-structure name CAN'T OPEN

The specified device cannot be OPENed; either because it does not exist or because the OPEN UUO failed. (MOUNT)

?CAN'T OPEN DEVICE dev

The specified device does not exist or it is assigned to another user. (DAEMON)

?CAN'T OPEN SWAP UNIT abc

DAEMON attempted to use the indicated swapping unit and failed. (DAEMON)

?CAN'T READ STRLST.SYS

A read error occurred when the system tried to read STRLST.SYS, the system file that contains necessary information about the structures known to the system. Consult the system administrator. (MOUNT)

?file structure name CAN'T REDEFINE FILE STRUCTURE n

"n" is the error code returned from the STRUUO which is attempting to redefine the structure to include the user. See the STRUUO UUO error codes in the DECsystem-10 Monitor Calls manual and in the UUOPRV specification. This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. Consult the system administrator. (MOUNT)

?file structure name CAN'T REMOVE FILE STRUCTURE FROM SEARCH LIST n

"n" is the error code returned from the STRUUO which is attempting to remove the structure from the user's search list. See STRUUO UUO error codes in the DECsystem-10 Monitor Calls manual and in the UUOPRV specification. This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. Consult the system administrator. (DISMOUNT)

?CAN'T RENAME-FILE PRESERVED

An attempt was made via the /DISPOSE:RENAME switch to delete a preserved file (i.e., a file whose owner's field is greater than 0). (QUEUE)

?CAN'T SET OUR SEARCH LIST

DAEMON tried to set its search list and failed in its attempt. (DAEMON)

?CAN'T SET SEARCH LIST = USER'S

DAEMON attempted to set its file structure search list to be the same as the user's search list. (DAEMON)

?CMLLRE (n) FILE WAS NOT FOUND - filename

The specified was not found on the user's directory.

?COMMAND ERROR

General catch-all error response for most commands. The syntax of the command is in error, and the command cannot be deciphered.

In FILCOM, one of the following errors occurred in the last command string typed.

1. There is no separator (- or =) between the output and input specifications.
2. The input specification is completely null.
3. The two input files are not separated by a comma.
4. A file descriptor consists of characters other than alphanumeric characters.
5. FILCOM does not recognize the specified switch.
6. The project-programmer number is not in standard format, i.e., [proj,prog].
7. The value of the specified switch is not octal.
8. The first input file is followed by a comma but the second input file is null.

?COMMAND SWITCH REQUIRED

The given command string requires a MAKLIB command code. (MAKLIB)

?COMMAND SYNTAX ERROR
TYPE /H FOR HELP

An illegal command string was entered. (GLOB)

n COMMANDS IN QUEUE

The request typed by you has been placed in a queue to be performed when possible. n is the number of requests in the queue for all users. (FILE, MOUNT, DISMOUNT)

?COMMA REQUIRED IN DIRECTORY

A project-programmer number has been specified without the separating comma. (DUMP, QUEUE)

CONT BY OPR

The job has been continued by the operator. This message appears on the console of the job being continued. (JCONT)

?CONTROL AND LOG FILES MUST BE DISTINCT

The control file cannot be the same file as the log file. (QUEUE)

?2K CORE NEEDED AND NOT AVAILABLE

FILCOM needs 2K of core to initialize I/O devices and this core is not available from the monitor. (FILCOM)

%CPUn OPR1 ACTION REQUESTED

The Job's CPU specification includes a CPU which is not running or is not scheduling jobs. The monitor remembers the specification and uses the CPU as soon as it is started. If at least one CPU is running, the message is printed only once, since the job can run on another CPU.

?DAEMON FILE MUST BE WRITTEN ON A DISK

The device specified was a nondisk device. (DAEMON)

?DAEMON NOT RUNNING

The DAEMON program has not been initialized. It must be started by the operator to allow the DUMP and DCORE commands to operate. (DUMP, DCORE)

?DETACH UUO FAILED

DAEMON could not detach itself from the TTY. Note that DAEMON does not detach itself if it is loaded with DDT. (DAEMON)

?DATA ERROR ON DEVICE PTR

A read error has occurred on the paper-tape reader.

?DESTINATION DEVICE ERROR

An I/O error occurred on the output device. (GLOB)

?DEVICE CAN'T BE REASSIGNED

1. The job's controlling terminal cannot be reassigned,

2. the logical name would be duplicated, or

3. the logical name is a physical device name and the job reassigning the device is either logged in under a different project-programmer number or is not the operator.

?DEVICE ERROR ON OUTPUT DEVICE

A write error has occurred on the output file. (MAKLIB)

?DEVICE INIT FAILURE

The specified device has been assigned to another job or does not exist.

?DEVICE MUST BE A DECTAPE

The only device that can be specified in the DTCOPY command string is the DECtape.

?DEVICE NOT ASSIGNABLE

A non-privileged user cannot assign the requested device because it belongs to the restricted pool of devices. The user should try to assign the device with the MOUNT command. (ASSIGN)

?DEVICE NOT AVAILABLE

Specified cannot be initialized because another user is using it or because it does not exist.

?DEVICE WILDCARD ILLEGAL

The wildcard construction cannot be used in the device specification. (DUMP, QUEUE)

?DIALOG MODE NOT SUPPORTED

The capability of interactive dialogue with the user has not been implemented. (QUEUE)

?DIRECTORY FULL ON OUTPUT DEVICE

There is no room in the file directory on the outut device to add the updated file (nondisk devices only). (MAKLIB)

device name DISMOUNTED

The DISMOUNT command has completed.

?device name DISMOUNT INCOMPLETE

The DISMOUNT command was unsuccessful. In most cases, the reasons for failure have already been listed by nonerror messages.

DON'T KNOW CITY LINE NUMBER

The DCORE command cannot be typed on CTY. (DAEMON)

?DOUBLE DEVICE ILLEGAL

Two device names appeared in a row without an intervening filename, or two colons appeared in a row, e.g., LPT:PTP: or DSKA ::FILEX. (DUMP, QUEUE)

?DOUBLE DIRECTORY ILLEGAL

Two directory names cannot appear without an intervening filename. (DUMP, QUEUE)

?DOUBLE EXTENSION ILLEGAL

Two extensions cannot appear without an intervening filename or comma. (DUMP, QUEUE)

?DOUBLE FILENAME ILLEGAL

Two filenames appeared in a row, or two periods appeared in a row; e.g., Q TEST1 TEST2 or TESTX.MAC. (DUMP, QUEUE)

?DPA n ILLEGAL UNIT

The user has specified a drive that does not exist in the system. (MOUNT)

%DRTLKE NON-EXISTENT FILE filename

A file was specified in the command string that is not present in the user's default directory.

?DSK CAN'T BE REASSIGNED

An attempt was made to reassign the prototype disk device data block (DDB).

?DSKCHR FAILURE n ON UNIT abc

The DSKCHR UUO gave an unexpected error return; n is the disk error code. Notify the operator. (DAEMON, KJOB)

?ENTER ERROR n
?DIRECTORY FULL

No additional files can be added to the directory of the output device; n is the disk error code. (GLOB)

?ENTER FAILURE

The DECtape directory is full (i.e., there is no room for the file to be written on the DECtape).

?ENTER FAILURE n

The output filename is null; n is the error code for an illegal filename (nondisk devices only). (MAKLIB)

?ENTER FAILURE IN QUEUE MANAGER

QUEUE was unable to enter the files into the output queue. (QUEUE)

?ENTER FAILURE n ON DAEMON FILE

The ENTER to write the file failed; n is the disk error code.

?ENTRY BLOCK TOO LARGE PROGRAM name

The entry block of the named program is too large for the MAKLIB entry table, which allows for 100 entry names. MAKLIB can be reassembled with a larger table. (MAKLIB)

?ERROR CLOSING OUTPUT, STATUS = n

An I/O error occurred while closing the file on disk; n is the disk error code (DUMP).

?ERROR IN JOB 0

Usually this message indicates an error has occurred in the monitor. The message appears on either the user's terminal or the operator's terminal and precedes a one-line description of the error.

?ERROR IN DETACHED JOB n [prog name]

A fatal error occurred in the detached job or in the monitor while servicing the job. This message appears only on the operator's terminal (OPR) and precedes a one-line description of the error.

?ERSATZ DEVICE ersatz device

Attempt was made to remove an ersatz device from the search list with R. (SETSRC)

?EXCEED LOG-OUT m QUOTA BY n BLOCKS

The total number of blocks for all the user's files exceeds the maximum permitted value (mmm) by the indicated amount (nnn). If you are the operator, the DISMOUNT proceeds. Otherwise, this message is followed by the message "?DISMOUNT INCOMPLETE". Then, you may use PIP or the DELETE command to remove files. But, until the number of blocks is under the limit, you cannot dismount a structure or log out. (DISMOUNT)

?EXECUTION DELETED

A program is prevented from being executed because of errors detected during assembly, compilation, or loading. Loading is performed, but the loader exits to the monitor without starting execution. (LOADER)

?EXPECTED FORMAT IS nnnK = 16K to 256K

The core-bank specified while processing the /T switch is not within the acceptable range or does not terminate with the letter K; e.g., 32 is not acceptable; 32K is. (COPY program)

file structure name FILE ERRORS EXIST

One of the files in a file structure has an error status, as flagged in the UFD of that file structure. (LOGIN)

?FILENAME ALREADY IN USE

The specified file already exists. (COMPIL)

?FILENAME REQUIRED FOR INPUT QUEUE

A file cannot be entered into the Batch input queue without a filename. (QUEUE)

?filename FILE NOT FOUND

File specified cannot be located.

?FILE n NOT IN SAV FORMAT

You indicated via the /X switch that the file is to be expanded but the specified file is not in compressed file format. n is either 1 or 2 indicating the first file or the second file. (FILCOM)

?FILE n READ ERROR

An error has occurred on either the first or second input device. (FILCOM)

?FILE REQUEST NOT QUEUED

Your file request was not queued. Other messages give the reasons. (FILE)

?FILE R ILLEGAL IF DSK: CONTAINS NO UFD

You tried to do a FILE R command with a search list that has no device that is eligible to receive the recalled files. You should have at least one structure in your active search list that contains both a UFD and the status of /CREATE and /WRITE. You should adjust your search list with the SETSRC command and retry the FILE command. (FILE)

%FILES ARE DIFFERENT

The two input files specified in the command string are different (i.e., the two files are not two versions of the same file but are two different files). (FILCOM)

?FILE SWITCHES ILLEGAL IN OUTPUT FILE

File switches cannot appear on the left of the equal sign, i.e., in the output specification. (QUEUE)

?(3) FILE WAS BEING MODIFIED-file.ext

Another user is modifying the file. (COMPIL)

?(0) FILE WAS NOT FOUND-file.ext

The named file could not be located. (COMPIL)

?FORMAT OR READ ERROR IN AUXACC.SYS

LOGIN unexpectedly found an end-of-file or an error in AUXACC.SYS. Notify the operator. (LOGIN)

file.ext FOUND BAD BY FAILSAFE READING MTA

The file in the file structure has an error status as flagged in the UFD of the file structure. (LOGIN)

FROM JOB n

An informative message telling you the job number to which the console was attached or from which the console is detaching. (ATTACH, DETACH)

?GIVING BACK TOO MUCH CORE

An internal problem in the DUMP program. Notify your system programmer or software specialist. (DUMP)

?HALT AT USER adr

Your program executed a HALT instruction at adr. Typing CONTINUE resumes execution at the effective address of the halt instruction.

file.ext HARDWARE DATA READ ERROR DETECTED

The file has a hardware data read error flagged in the UFD of the file structure. (LOGIN)

file.ext HARDWARE DATA WRITE ERROR DETECTED

The file has a hardware data write error flagged in the UFD of the file structure. (LOGIN)

%structure name HAS OTHER USERS-/REMOVE IGNORED

The specified structure is still being used by other jobs. The structure is removed from your search list but is not physically dismounted. (DISMOUNT)

?HAVE OPERATOR START OMOUNT, THEN TRY AGAIN

The command area where your requests are stored does not exist. (FILE, MOUNT, DISMOUNT)

?HUNG DEVICE dev

If a device does not respond within a certain period after it is referenced, the system decides that the device is not functioning and outputs this message.

?ILLEGAL BLOCK TYPE dev:file.ext

The block type used is not in the range 0-77. (MAKLIB)

?ILLEGAL BLOCK TYPE

While loading the bootstrap loader with the /L switch, an unrecognizable block type was encountered by DTCOPY. (DTCOPY program)

?ILLEGAL COMMAND SYNTAX CHARACTER x

The character x is used incorrectly in the command string. (QUEUE)

?ILLEGAL DATA MODE FOR dev

The data mode specified for a device in your program is illegal, such as dump mode for the terminal. (MAKLIB)

?ILLEGAL DEVICE - dev:

you specified a file structure name that is not a disk. (FILE)

drive ILLEGAL DRIVE NAME

The drive requested by the user is not one recognized by the system. (MOUNT)

?drive ILLEGAL DRIVE NAME

The drive specified by the user is in conflict with the unit or controller type required by the units of the file structure. (MOUNT)

?(0) ILLEGAL FILENAME

A filename of zero was specified. (MAKLIB)

?ILLEGAL IN BATCH JOB

The ATTACH, DETACH, SEND, CCONT, and CSTART monitor commands cannot be used by a batch job.

?ILLEGAL JOB NUMBER

The job number is too large or is not defined in this configuration.

?ILLEGAL NAME - name

You specified a structure name that does not exist. (MOUNT, DISMOUNT)

?ILLEGAL QUEUE DEVICE

The queue name specified cannot be used with the given switch. (QUEUE)

?ILLEGAL QUEUE NAME xxx

The queue is not of the system queues, or the queue is a logical name. (QUEUE)

?ILLEGAL SWITCH

A non-recognizable switch was used in the command string. (GLOB)

?ILLEGAL TO CREATE REQUEST FOR SOMEONE ELSE

Only the operator logged in under 1,2 can create queuing request for other users. (QUEUE)

?drive name ILLEGAL UNIT

The drive specified by the user is not a valid drive name. (MOUNT, DISMOUNT)

?ILLEGAL UUO AT USER adr

An illegal UUO was executed at user location adr.

?ILL INST. AT USER adr

An illegal operation code was encountered in your program.

?ILL MEM REF AT USER adr

An illegal memory reference was made by your program. If this message occurred on a memory write, the error is at adr-1 since the program counter has been advanced. If it occurred on a memory read, then the illegal instruction is probably in location adr. You should use the E command to first examine location adr-1 and then location adr in order to determine the illegal instruction. The index registers may also have to be examined.

?INCORRECT VALUE FOR /WRITE SWITCH:xxx

You specified the value "xxx" with the /WRITE switch. The only acceptable values are YES and NO. (MOUNT)

?INPUT AND OUTPUT DECTAPES MAY NOT BE THE SAME DEVICE

The COPY program prforms its operations on an input DECtape and an output DECtape. These DECtapes cannot be the same. (DTCOPY program)

?INPUT (or OUTPUT) BLOCK TOO LARGE

A DECtape block number greater than 1101 (octal) was encountered. (DTCOPY program)

?INPUT (or OUTPUT) CHECKSUM OR PARITY ERROR

A read (or write) error has been detected. (DTCOPY program)

?INPUT DEVICE dev CANNOT DO OUTPUT AT USER adr

Output was attempted on a device that can only do input (e.g., the card reader).

?INPUT (or OUTPUT) DEVICE ERROR

The DECtape control unit has detected the loss of data or a missed block. (DTCOPY program)

?INPUT DEVICE NOT A DISK

The input specifications in a QUEUE command must be disk files. (QUEUE)

?INPUT ERROR

An I/O error occurred while reading a temporary command file from the disk. File should be rewritten. (COMPIL)

?INPUT ERROR - file.ext FILE NOT FOUND

The specified file could not be found on the input device. (FILCOM)

?INPUT ERROR ON DEVICE dev: STATUS (nnnnnn)

A data or device error occurred on input. (MAKLIB)

?INPUT ERROR, STATUS = n

An I/O error occurred while reading the file from disk; n is the disk error code. A new INPUT command causes a new LOOKUP to be done. (DUMP, DAEMON)

?INPUT (or OUTPUT) PREMATURE END OF FILE

When copying a DECtape, DTCOPY encountered the end of file before it expected it. This may happen when copying a PDP-9 DECtape to a DECsystem-10 DECtape. (DTCOPY program)

?INSUFFICIENT CORE FOR QUEUE

There is no enough core in system at the time of the KJOB command to make an output queue entry. (QUEUE)

?INVALID ENTRY - TRY AGAIN
#

An illegal project-programmer number or password was entered and did not match identification in system. The user is to retype his project-programmer number and password. (LOGIN)

?I/O TO UNASSIGNED CHANNEL AT USER adr

An attempt was made to do an OUTPUT, INPUT, OUT, or IN to a device that the user's program has not initialized.

?x IS AN ILLEGAL CHARACTER SWITCH

An illegal character or switch was encountered in the command string. (MAKLIB)

?symbol IS A MULTIPLY DEFINED LOCAL

The named symbol is in more than one symbol table with different values. (DUMP)

?symbol IS AN UNDEFINED SYMBOL

The named symbol is not in DUMP's symbol table. (DUMP)

?symbol IS AN UNDEFINED SYMBOL TABLE NAME

The named symbol table has not been loaded with an XTRACT command. (DUMP)

?JOB CAPACITY EXCEEDED

1. This message is received by a user who attempts to login after the maximum number of jobs that the system has been set to handle has been initiated. The user should login at a later time. (LOGIN)

2. This message is received when a system program must create a job in order to perform its operation and the maximum number of jobs has been initiated. (SYSTAT, HELP, ATTACH)

3. This message may appear on rare occasions when an error made by the system hardware causes a reduction in the job capacity of the system.

?JOB NOT WAITING

The job specified is not waiting to be continued. (JCONT)

JOB SAVED

The output is completed.

```
JOBn USER [p,p] LOGGED OFF TTY n AT hhmm dd-mm-yy
DELETED <ALL> n FILES
SAVED <ALL> n FILES m TOTAL BLOCKS USED
ANOTHER JOB STILL LOGGED IN UNDER [p,p]
RUNTIME n MIN m SEC
```

This information is typed as you log off successfully. Note that m is total blocks allocated as opposed to blocks written. Therefore, it is always greater than or equal to the number of blocks written. Files are allocated in units of blocks called clusters. The system administrator selects the cluster size for each file structure, usually one block per cluster for FH file structures, and 5 or 10 blocks per cluster for DP file structures. (KJOB)

?LANGUAGE PROCESSOR CONFLICT

The use of the + construction has resulted in a mixture of source languages. (COMPIL)

?LEVEL D ONLY

The command issued is available only in 5-series monitors.

[LGNQTA structure IN: iiiii OUT: oooooo USED; nnn]

The monitor has computed the disk usage on structure to be iiiii input file quota, oooooo output file quota, and nnn blocks used.

[LGNRDU RECOMPUTING DISK USAGE]

The monitor is recomputing the disk usage on the default directory, before the user can successfully LOGIN.

?LGTISK ILLEGAL SYNTAX IN KJOB COMMAND

An illegal command syntax has been specified for a command to KJOB. (LOGOUT)

?LGTCSK CONFLICTING SWITCHES IN KJOB COMMAND

Conflicting switch-types have been specified to the KJOB command. (LOGOUT)

%LGTURS UNRECOGNIZED SWITCH switch - IGNORED

An illegal (unrecognizable) switch has been specified for KJOB, it is ignored. (LOGOUT)

%LGTJWF JOBSTR UUO FAILED - NO QUOTA ENFORCED

The JOBSTR monitor call was executed, and it failed; therefore, no quotas are in effect. (LOGOUT)

%LGTCCW CAN'T CLEAR SOFTWARE WRITE-LOCK ON STRUCTURE structure

You had structure software write-locked, and the STRUUO to clear this condition failed. This message is usually found by a %LGTURF message. (LOGOUT)

[LGTWFI WAITING FOR structure UFD INTERLOCK]

LOGOUT attempts to set the monitor UFD interlock before doing quota enforcement. If another job has it interlocked already, LOGOUT will wait. (LOGOUT)

[LGTRDU RECOMPUTING DISK USAGE ON structure]

Disk usage is being recomputed on the specified structure. (LOGOUT)

?LGTOUF OPEN UUO FAILED FOR structure

The specified structure tried to execute an OPEN monitor call, but it failed. (LOGOUT)

?LGTLQE structure LOGGED-OUT QUOTA nnn EXCEEDED BY nn BLOCKS

Your allocation on the file structure named is greater than your logged out quota. The user must go through the CONFIRM dialogue and delete files until you are under the quota allowed to log off. (KJOB, LOGOUT)

%LGTULF ZSTRUCTURE UFD LOOKUP FAILURE nn

A LOOKUP was executed on the specified structure, and it failed.

%LGTURF structure UFD RENAME FAILURE nn

A RENAME was executed on the specified structure, and it failed.

```
                                                   DAEMON NOT RUNNING
?LGTASF ACCOUNTING SYSTEM FAILURE DAEMON UUO FAILED
                                                   ERROR CODE NO.
```

An error occurred in the system accounting files, and the DAEMON UUO failed. Call the operator if you receive such a message. If the operator receives the message, continue the system for system debugging.

?LINKAGE ERROR - RUN UUO

An I/O error occurred while reading a program from the device SYS:. (COMPIL)

%LISTING DEVICE OUTPUT ERROR, STATUS =

The device specified for the output has an error. A new OUT command selecting a new file can be given or an OUT and APPEND command sequence to try again. (DUMP)

?LISTING ENTER FAILURE n

The ENTER to write the output file failed; n is the disk error code. (QUEUE)

?LISTING OPEN FAILURE ON DEVICE dev

The OPEN failed on device dev. (QUEUE)

?file structure LOCKED OUT BY OPERATOR

The operator has locked this structure, preventing any new accesses, in order that it may be removed. (MOUNT)

%LOG FILE DEVICE IS NOT DSK OR SPOOLED; WILL USE TTY

The log file device is not a disk or spooled; therefore, KJOB assumes TTY. (KJOB)

file structure name LOGGED OUT QUOTA n EXCEEDED BY m BLOCKS

Your allocation on the file structure named is greater than your logged out quota. You must go through the CONFIRM dialogue and delete files until you are under the quota allowed to log off. (KJOB, LOGOUT)

%LOGICAL NAME WAS IN USE, DEVICE dev ASSIGNED

You previously assigned this logical name to another device. The logical name is cleared from the first device and assigned to the second.

LOGICAL NAME IN USE (NAME IGNORED)

You specified a logical name in your MOUNT command that is currently in use by the same job. MOUNT creates a unique logical name of the form "Mnnnnn" for this MOUNT request. (MOUNT)

?LOGIN PLEASE

A command that requires you to be logged in has been typed to the monitor; it cannot be accepted until your perform a LOGIN.

?LOGIN PLEASE TO USE SWITCH CREATE

You must be logged in to make a new entry into a system queue. (QUEUE)

?LOOKUP ERROR n
?file.ext FILE NOT FOUND

The named file cannot be found in the directory on the specified device. (GLOB)

?LOOKUP FAILED, 'BSLDR.REL'

While processing the /L switch, COPY could not find the bootstrap loader named BSLDR.REL. (COPY program)

?LOOKUP FAILURE

The LOOKUP to read the disk file failed. This message is followed by a line explaining the reason for failure. (MAKLIB)

?file structure name LOOKUP FAILURE n

The LOOKUP to read the file failed; n is the disk error code.

LOOKUP FAILURE FOR AUXACC.SYS

The system was unable to find AUXACC.SYS (one of the system's master accounting files). The system is unable to determine quotas on your file structure. For MOUNT commands this means that no UFD can be created. (MOUNT, DISMOUNT)

?LOOKUP FAILURE FOR INPUT FILE n file

DUMP cannot read the input file. (DUMP)

?LOOKUP FAILURE n ON DAEMON FILE

The LOOKUP to read the DAEMON file failed; n is the disk error code. (DAEMON)

%LOOKUP FOR HOME BLOCKS FAILED

The system is unable to locate or access HOME.SYS on the mounted structure. Your job will continue, this allowing unformatted structures to be accessed. (MOUNT)

?MAKLIB SYNTAX ERROR

An illegal command string was entered; for example, the equal sign was omitted or a program name was specified for the output file. (MAKLIB)

?MAX = n

A value was specified for an argument that is greater than the maximum value (n) allowed. (DUMP)

?MAY NOT LOGIN AS MFD PPN

No one can login as [1,1] because this number is the project-programmer number of the MFD. (LOGIN)

```
                LOCAL
                REMOTE
?MAY NOT LOGIN  DATA SET
                BATCH JOB SUBJOB
                REMOTE CTY
                OPR
```

ACCT.SYS entry does not permit the project-programmer number to login at the terminal that is being used. (LOGIN)

?MAY NOT LOGOUT WITH FILE STRUCTURES FOR LOGICAL NAMES

A file structure in the job's search list is assigned a logical name, and only physical device names are recognized. The user should deassign the logical names. (KJOB, LOGOUT)

?MEM PAR ERROR AT USER PC adr

The processor detected a memory parity error in the low or high segment while the job was executing. The adr is the address of the PC stored by the hardware rather than the user address of the parity error. The operator also receives an error message giving the range of absolute addresses in case memory reconfiguration is necessary. DAEMON is awakened in order to record the pertinent information about the error for field service personnel.

You must start a new copy of your program by typing the appropriate monitor command R, RUN, or GET. You should not start the program over by typing START, since the error is likely to reoccur or the program operate with incorrect data.

?MKLBTL ENTRY BLOCK TOO LARGE, PROGRAM name

The entry block of the named program is too large for the MAKLIB entry table, which allows for 1000 entry names. MAKLIB can be reassembled with a larger table. (MAKLIB)

?MKLDFO DIRECTORY FULL ON OUPUT DEVICE

There is no room in the file directory on the output device to add the updated (non-disk devices only). (MAKLIB)

?MKLDNA dev NOT AVAILABLE

The specified device does not exist or is assigned to another user. (MAKLIB)

?MKLEFI ENTER FAILURE

The ENTER to write the disk file failed. This message is followed by a line explaining the reason for the failure. (MAKLIB)

?MKLEFO ENTER FAILURE n

The output filename is null; n is the error code for an illegal filename (non-disk devices only). (MAKLIB)

?MKLEOD DEVICE ERROR ON OUTPUT DEVICE

A writer error has occurred on the output file. (MAKLIB)

?MKLIBT ILLEGAL BLOCK TYPE dev:file.ext

The block type used is not in the range 0-77, 1000-1777, or ASCII text from .TEXT pseudo-op. (MAKLIB)

?MKLIDM ILLEGAL DATA MODE FOR dev

The data mode specified for a device in the user's program is illegal, such as dump mode for the terminal. (MAKLIB)

?MKLIFN (0) ILLEGAL FILENAME

A filename of zero was specified. (MAKLIB)

?MKLMSE MAKLIB SYNTAX ERROR

An illegal command string was entered; for example, the left arrow was omitted or a program name was specified for the output file. (MAKLIB)

?MKLNEA NOT ENOUGH ARGUMENTS

An insufficient number of files of one type has been specified. (MAKLIB).

%MKLNIN WARNING NO INDEX ON OUPUT FILE-CONTINUING

The user has changed the structure of the indexed library file when deleting, appending, or inserting, thereby invalidating the index. The index has been removed from the new file. Reindexing is required. (MAKLIB)

?MKLNIO CANNOT DO I/O AS REQUESTED

Input (or output) cannot be performed on one of the devices specified for input (output). For example, input may have been requested for a device that can only do output. (MAKLIB)

?MKLNPS dev:file.ext/switch:( ) NO PROGRAM NAME SPECIFIED

The switch (/D or /R) used in the command string requires that a program name be given. (MAKLIB)

?MKLTEI TRANSMISSION ERROR ON INPUT DEVICE dev

A transmission error has occurred while reading data from the specified device. (MAKLIB)

?MKLTMN TOO MANY FILENAMES OR PROGRAM NAMES

More than 40 program names or filenames were specified in the command string. The user should separate the job into several segments. (MAKLIB)

?MKLUMT UNEQUAL AMOUNT OF MASTER AND TRANSACTION PROGRAMS

On a replace request, the number of master programs (or files) does not equal the number of transaction program (or files). (MAKLIB)

MONITOR MUST BE 5.03 or later

This version of UMOUNT must be run on a 5.03 or later monitor. (FILE, MOUNT, DISMOUNT)

?MORE THAN ONE OUTPUT FILE ILLEGAL

Only one output queue-name may be specified in the QUEUE command string. (QUEUE)

?device MOUNTED

The device is mounted and ready for use. The MOUNT command has completed. If a file structure was mounted, a list of the unit ID's and the drives on which they are mounted is output. (MOUNT)

?MOUNT INCOMPLETE

UMOUNT was unable to complete the user's MOUNT request. Other messages give the reasons. (MOUNT)

?MOUNT UNSUCCESSFUL

UMOUNT was unable to process the MOUNT request. Other messages accompany this one to give the reasons. (MOUNT)

?MUST BE IN OWNER'S PROJECT TO REQUEST SINGLE ACCESS

You may not request single-access (/SINGLE switch) unless you have the same project number as the owner of the file structure. This requirement is enforced since a user with single access may execute super-USETI/USETO UUOs. (MOUNT)

?MUST BE LOGGED IN

You must be logged in to queue MOUNT, DISMOUNT, and FILE requests. (Note that with a change to the monitor, these commands may be used with the check or HELP functions without you being logged in.) (MOUNT)

NAME:

The ACCT.SYS entry for this project-programmer number requires you to type a name which matches the one in ACCT.SYS in order to login. (LOGIN)

?NEED 5.03 OR LATER FOR REATTACH COMMAND

The REATTA program depends on UUOs available in the 5.03 release of the monitor. The user attempted to run the program using an older monitor. (REATTA)

?NESTING TOO DEEP

The @ construction exceeds a depth of nine and may be due to a loop a @ command files (COMPIL).

?NETUNN - UNDEFINED NETWORK NODE

A node/station has been specified that is not in contact with the command decoder.

NO COMMANDS IN QUEUE

There are no requests (of the specified type; MOUNT, DISMOUNT, or FILE) in the queue from any of the users. This message may be issued in response to any one of the following commands: MOUNT /CHECK, DISMOUNT /CHECK, or FILE C. (FILE, MOUNT, DISMOUNT)

?NO CORE ASSIGNED

No core was allocated when the GET command was given and no core argument was specified in the GET.

%NO DEVICE FOR DIRECTORY

A FILE command has been issued and you have no device in your active search list that is eligible to have the directory written on it. That is, the search list does not contain a structure that has both a UFD for you and a status of /CREATE and /WRITE. The command is processed but no directory is created. (When UMOUNT writes directories, it attempts to put them on the first structure in your search list that has both a UFD and a status of /CREATE and /WRITE.) (FILE)

NO DIFFERENCES ENCOUNTERED

No differences were found between the two input files. (FILCOM)

?(1) NO DIRECTORY FOR PROJECT-PROGRAMMER NUMBER - file.ext

A UFD does not exist for the requested project-programmer number. (COMPIL)

?NO END BLOCK ENCOUNTERED

The last block of the bootstrap loader program must be an end block (refer to the MACRO manual). (DTCOPY program)

?NO ENTRY IN AUXACC.SYS
NO SEARCH LIST OR UFDS CREATED

If you have no entry in AUXACC.SYS, LOGIN does not create UFDS or a search list. You are logged-in and have UFDs if they existed previously. You may write only on file structures that have UFDs or read all file structures. You may also create a file structure search list with SETSRC. You can create UFDs on those file structures for which you have an entry in QUOTA.SYS by using the MOUNT command. (LOGIN)

NO ENTRY IN QUOTA.SYS OR AUXACC.SYS

You do not have a UFD on the specified structure. Your ability to access and create files on the structure depends on the individual protection of the structure's other UFDs and files. (MOUNT)

%NO INFO ON name

You specified a feature that has no available documentation. (HELP)

?NO MODIFIER ALLOWED IN SWITCH switch

The switch specified cannot have an argument. (QUEUE)

NONE PENDING FOR YOUR JOB

You have specified the WAIT or CHECK option in your FILE, MOUNT, or DISMOUNT command and there are no pending requests in the queue. (FILE, MOUNT, DISMOUNT)

?NON-EXISTENT DRIVE DPAm

The user has specified a drive that does not exist in the system. (MOUNT)

%NON-EXISTENT FILE input specification

The file specified for input could not be found. This message is not output if the /NEW switch is specified for the file. (QUEUE)

?NON-EX MEM AT USER adr

Usually due to an error in the monitor.

?NO OPR.JOB FOR THIS REQUEST

An operator request has been issued, but there is no OMOUNT running and enabled to service the request. The request is still queued unless the /PAUSE switch was given.

?NO PRIVILEGES TO SET CPU

You do not have the privilege set by LOGIN from ACCT.SYS to change the CPU specification. You should request that these privilege bits be set by the system manager.

?NO PRIVS TO SET PRIORITY THAT HIGH

You do not have privileges to set such a high priority.

?NO PRIVS TO UNSPOOL

You do not have privileges to unspool devices, and the operator has not set bit 28 in the STATES word.

?dev:file.ext < NO PROGRAM NAME SPECIFIED >

The switch (/D or /R) used in the command string requires that a program name be given. (MAKLIB)

?NO REMOTE USERS. TRY AGAIN LATER

The operator has used the SET SCHEDULE command to prevent LOGINs from remote terminals. The message of the day is still typed. (LOGIN)

NO ROOM IN QUEUE, TRY AGAIN LATER

There is no room in the queue for your request to be sent to the operator. (MOUNT)

?NO ROOM IN QUEUE, TRY AGAIN LATER

The [3,3] command area (where your requests are stored) is full. (FILE, MOUNT, DISMOUNT)

?(14) NO ROOM OR QUOTA EXCEEDED - file.ext

There is no room on the file structure or your quota on the file structure has been exceeded.

%NO RUNNING CPUS IN SPECIFICATION

If none of the CPUs in the job's CPU specification are running, the user receives this message every minute until the CPU is started or he types a new SET CPU command.

?NO START ADR

Starting address or reenter address is zero, because the user failed to specify the starting address in the END statement of the source program or in the START command. However, an implicit starting address of 0 may be specified.

?NO STRUCTURE NAME

You did not specify a device in your DISMOUNT command. (DISMOUNT)

?NO SUCH DEVICE

The device name does not exist or was not assigned to this job.

?NO SUCH JOB

An attempt was made to attach to a job that has not been initialized.

?NO SUCH STR

A nonexistent file structure was specified. (KJOB)

?NO SUCH TTY

The terminal number is not part of the system configuration.

?NO SUCH UNIT

The unit does not exist or all units of this type are in use.

?NO SYS

This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. (FILE, MOUNT, DISMOUNT)

?NOT A JOB

The job number is not assigned to any currently running job. (ATTACH, DSK, JCONT). There is no job logged in at this terminal. (CONTINUE)

?NOT A SAVE FILE

The file is not a core image file.

?NOT A SPOOLING DEVICE

The device specified is not one of the spooling devices (LPT, CDP, CDR, PTP, PLT).

?NOT A STR - TRY AGAIN

The file structure specified is not recognized by the monitor.

?NOT A TTY

The device name given is not a terminal. (REATTA)

?NOT AVAILABLE

The operator has restricted use of the specified feature, or the console switches are set to disallow the use of a programmable address break.

?drive NOT AVAILABLE

The drive indicated by you is not currently available. (MOUNT)

?command NOT CODED

A command that is not in this version of DUMP was specified in the command string. (DUMP)

?NOT ENOUGH ARGUMENTS

An insufficient number of files of one type has been specified. (MAKLIB)

?NOT ENOUGH CORE

The system cannot supply enough core to use as buffers or to read in a system program. (COMPIL)

NOT ENOUGH DRIVES AVAILABLE

There are currently not enough drives of the right type to mount the file structure. (MOUNT)

NOT ENOUGH TABLE SPACE FOR SWAPPING UNITS

There are more swapping units than DAEMON allowed for. DAEMON should be reassembled. (DAEMON)

?dev file.ext program NOT FOUND

The file or the program was not found on the device or in the file specified. If a program name is printed, this message may indicate that the program names in the command string appear in a sequence different from their sequence within the file. Therefore, the program may actually exist but missed because of the incorrect sequence in the command string. (MAKLIB)

?file.SAV NOT FOUND

The program file requested cannot be found on the system device or the specified device.

unit-id NOT MOUNTED

The file structure is mounted but the specified unit is not. According to STRLST.SYS, the unit should belong to the structure. Consult the system administrator. This message is followed by the message "REQUEST QUEUED TO READY DRIVES." (MOUNT)

drive NOT READY

The indicated drive is either off-line or physically write-locked when write-enabled was requested. The operator will be notified. (MOUNT) unit-id NOT READY

The requested file structure exists but the unit is not ready. This message is followed by the message "REQUEST QUEUED TO READY DRIVES." (MOUNT)

?NOT YET SUPPORTED COMMAND CODE switch

A switch has been specified that is not implemented. (QUEUE)

NO UFD CREATED

You cannot have a UFD on this structure. Your ability to access and create files on the structure depends on the individual protection of the structure's other UFDs and files. (MOUNT)

NO UFD CREATED

You may access the file structure, but you cannot write in your disk area since you have no UFD. (MOUNT)

?NULL DEVICE ILLEGAL

A colon has been found without a preceding device name. (QUEUE)

?NXM adr

While computing the value of an expression, a non-existent location was specified when referencing the input file. (DUMP)

?nK OF CORE NEEDED or ?nP OF CORE NEEDED

There is insufficient free core to load the files; n is the size being requested for the segment that failed (either high or low segment, not the sum of the high and low segments). This message occurs when the virtual core for the system has been exceeded or the core for this job has been executed. You should type CORE, to determine what core has been exceeded, and whether the high or low segment was too big. K denotes 1024 words which is the unit of core allocation on a KA10-based system, and P denotes 512 words (one page) which is the unit of allocation on a KI10-based system.

?OFFSET = 1000 TO 777600 (OCTAL)

The offset specified by the user is not within the acceptable range. (DTCOPY program)

?ONLY BATCH USERS MAY LOGIN. TRY AGAIN LATER

The operator has used the SET SCHEDULE command to prevent LOGINs, except for BATCH jobs. The message of the day is still typed. (LOGIN)

?OPEN FAILURE ON DATA DEVICE dev

The OPEN on the specified device failed. (DUMP)

OPERATOR BUSY, HANG ON PLEASE.

The user must wait for the operator to become available.

OPERATOR NOTIFIED

1. The operator is available and the user may continue typing his message. (PLEASE)

2. A request is queued to the operator to perform a specified action. (MOUNT, DISMOUNT)

OPERATOR NOT ON DUTY

SCHED 400 is set (i.e., there is no operator on duty to handle MOUNT and DISMOUNT requests). If you did not specify the /SHOVE switch in your request, then the MOUNT or DISMOUNT cannot be completed. If you did specify the /SHOVE switch then you receive the message, "SHOVING REQUEST ANYWAY..." and normal processing resumes. (FILE, MOUNT, DISMOUNT)

OPERATOR NOT ON DUTY - /REMOVE IGNORED

The specified structure will be removed from your search list but will not be physically removed from the drive. (DISMOUNT)

OTHER USER - CANNOT SINGLE ACCESS

You issued a MOUNT command with the /SINGLE switch specified and others have use of the drive. The MOUNT will proceed and the /SINGLE switch will be ignored. (MOUNT)

OTHER USERS SAME PPN

A program has determined that other jobs are currently logged-in under the same project-programmer number. (LOGIN, KJOB)

?OUT OF BOUNDS

The specified adr is not in your core area, or the high segmenet is write-protected and you do not have privileges to the file that initialized the high segment. (D,E)

?OUTPUT DEVICE dev CANNOT DO INPUT AT USER adr

An attempt was made to input from an output device (e.g., the line printer).

?OUTPUT DEVICE ERROR

An error has occurred on the ouput device. (FILCOM)

?OUTPUT ERROR

An I/O error occurred while writing a temporary command file on disk. (COMPIL)

?OUTPUT ERROR, STATUS = n

An I/O error occurred while writing the file on disk; n is the disk error code. (DAEMON)

?OUTPUT INITIALIZATION ERROR

The output device cannot be initialized for one of the following reasons:

1. The device does not exist or is assigned to another job.

2. The device is not an output device.

3. The file cannot be placed on the output device. (FILCOM)

PASSWORD:

You must type a PASSWORD which matches that in the ACCT.SYS entry for this project-programmer number. Echoing is suppressed to preserve PASSWORD security. If the user is at a half-duplex (local copy) terminal, this message is replaced by a sequence of random over-typed characters, over which the user types his PASSWORD. (LOGIN)

PAUSE...(^C TO QUIT, CR TO CONT)

The /PAUSE switch has been specified, and an operator action is about to be requested. ^C aborts the command before the request is queued to the operator. Carriage return-line feed allows the command to continue, and the request is queued to the operator. (DISMOUNT)

?PC OUT OF BOUNDS AT USER adr

An illegal transfer has been made by the user program to user location adr.

?PLEASE KJOB OR DETACH

Attempt was made to LOGIN a job when you already had a job initialized at that terminal. (LOGIN)

?PLEASE LOGIN AS [OPR]

The operator is the only person that can initialize DAEMON by typing R DAEMON.

?PLEASE TYPE ^C FIRST

A command which would start a job has been issued after a CSTART or CCONT.

?PPN HAS EXPIRED

The current date is greater than the expiration date of the project-programmer number. The user may not login until expiration date is changed by the system manager. (LOGIN)

?PPN OR DEVICE PERMITTED ONLY ON FILE F OR FILE Z CMDS.

You specified a project-programmer number or a device with an implied project-programmer number (e.g., NEW:) in a FILE command that does not permit specification of a project-programmer number. A project-programmer number may be specified only on the FILE F and FILE Z commands. (FILE)

?PROGRAM ERROR WHILE RESETTING MASTER DEVICE

FUDGE2 cannot find the master device or cannot find the program on the master device. (FUDGE2)

?PROGRAM NOT PRIVILEGED - IGNORED!

UMOUNT must be run from SYS: or NEW:, or the user must be [1,2] in order for the user to queue requests. (FILE, MOUNT, DISMOUNT)

?PROJECT 1 MAY NOT BE PTY

Project 1 is never allowed to login over a pseudo-TTY. (LOGIN)

?PROTECTION FAILURE file-specification

The user does not have privileges to access the indicated file (file specification). (FILE)

?(2) PROTECTION FAILURE - file.ext

There was a protection failure or the directory on DECtape had no room for the file. (COMPIL)

?PTR INIT FAILURE

The logical device PTR is not available or could not otherwise be initialized. (DTCOPY program)

?QUENFI NO FILES IN REQUEST

An illegal queue request was made; the monitor aborted the request.

QUOTA.SYS LOOKUP FAILURE

The LOOKUP to read QUOTA.SYS failed. (MOUNT)

QUOTA.SYS NOT ON STRUCTURE

QUOTA.SYS is not part of this structure. The user may still use the file structure, but no UFD will be created. (MOUNT)

QUOTA.SYS READ ERROR

An I/O error occurred while reading QUOTA.SYS. (MOUNT)

QUOTA.SYS OR AUXACC.SYS READ ERROR

An I/O error occurred while QUOTA.SYS or AUXACC.SYS was being read. For MOUNT commands this means that no UFD can be created. (MOUNT)

QUOTA.SYS WRONG FORMAT VERSION

Wrong version of QUOTA.SYS is on the file structure being mounted. Consult the operator. (MOUNT)

%REASSIGN ILLEGAL FOR RESTRICTED DEVICE - /RELEASE ASSUMED

The user may not reassign a restricted device. If he attempts to do so, the device is released from his job and returned to the pool of restricted devices. (DISMOUNT)

%REASSIGN JOB # NOT ASSIGNED - /RELEASE ASSUMED

The user attemped to reassign a device to a job that doesn't exist. The device is released from his job. (DISMOUNT)

?/REELID REQUIRES MAGTAPE

The user specified a /REELID switch but the device he requested to be mounted is not a magnetic tape. The /REELID switch may be specified only when mounting magnetic tapes. (MOUNT)

?file structure name RENAME FAILURE n

The RENAME to change the protection of the file failed; n is the disk error code. (KJOB, LOGOUT)

?(4) RENAME FILENAME ALREADY EXISTS - file.ext

The new filename on a RENAME command already exists. (COMPIL)

REQUEST QUEUED

Your request to perform a specified action has been queued to the operator. (FILE, DISMOUNT, and MOUNT for non-file structures).

REQUEST QUEUED TO MOUNT UNITS

A request is queued to the operator to mount and ready the packs on the proper drives. (MOUNT)

REQUEST QUEUED TO READY DRIVES

The requested structure is mounted but one or more drives (as specified by previous messages) are not ready. A request is queued to the operator. (MOUNT)

?REQUIRES DEVICE NAME

The device name or file structure name is required with the MOUNT and DISMOUNT commands.

?REQUIRES "/REELID" SWITCH

You must include the /REELID switch and its argument in the command string when you want to have a magnetic tape mounted. (MOUNT for magnetic tapes)

?REQUIRES /VID SWITCH

You must include the /VID switch and its argument in your MOUNT command for DECtapes and for all input-only devices. (MOUNT for DECtapes and all input-only devices.)

%RESPONSE MUST BE SINGLE LETTER WITH OPTIONAL STR LIST

You have not followed the correct format for a response to CONFIRM:. Responses are limited to a single letter or a single switch, optionally followed by a space or tab and a list of file structures. A new response will be requested. (KJOB)

?RIGHT BRACKET REQUIRED IN DIRECTORY

The project-programmer number must be enclosed in square brackets. (QUEUE)

?SCNDFN DOUBLE FILE NAME ILLEGAL

A double file name has been specified (sometimes through use of *).

?SCNNSF NULL SFD ILLEGAL

A null SFD has been specified in the command string, and its use is illegal.

?SCNSVR SWITCH VALUE REQUIRED ON SINCE

A /SINCE was specified with no argument.

?SCNUDN UNRECOGNIZED NAME IN DATE/TIME

An illegal character appears after a time/date - type switch. (e.g., /SINCE:L-APR-L971.)

?SCNUKS UNKNOWN SWITCH switch

An unknown switch has been specified in the command string.

%SEARCH LIST DOES NOT ALLOW CREATES

There are no file structures available to the user on which he can write. Run MOUNT or SETSRC to modify the search list as necessary. (LOGIN)

%SEARCH LIST IS EMPTY

There are no file structures in the DSK: search list that are available to the user. He can run the SETSRC program to modify his search list. (LOGIN)

SHOVING REQUEST ANYWAY...

SCHED 400 is set (i.e., there is no operator on duty to handle MOUNT and DISMOUNT requests). However, the user has specified the /SHOVE switch so the request will be queued even though there may be no one at the computer to MOUNT or DISMOUNT the device(s). (FILE, MOUNT, DISMOUNT)

?SINGLE ACCESS BY JOB n

You requested to have a structure mounted that is already mounted by job n with the /SINGLE switch specified. (FILE, MOUNT)

file.ext SOFTWARE CHECKSUM OR REDUNDANCY ERROR

The file has no error as flagged in the UFD of the file structure. (LOGIN)

?SOME OTHER TIME

You are not scheduled to LOGIN at this time. You should try again when you are allowed to login. (LOGIN)

?SORRY, CAN'T OPEN DSK, PLEASE CALL THE OPERATOR

This message is returned from the GRIPE program.

?SORRY, CAN'T WRITE IN COMPLAINT AREA, PLEASE CALL THE OPERATOR

This message is returned from the GRIPE program.

?SORRY, COMPLAINT BASKET IS FULL, PLEASE CALL THE OPERATOR

This message is returned from the GRIPE program.

?SORRY, NO UFD FOR COMPLAINT BASKET, PLEASE CALL THE OPERATOR

This message is returned from the GRIPE program.

%SPECIFIED LOGICAL NAME WAS IN USE

You specified a logical name in your MOUNT command that was already in use by the same job. The device with which the logical name was previously associated is no longer associated with the logical name. The logical name is now associated with the newly mounted device. (MOUNT)

?STATION NUMBER INVALID

The requested station number is not recognized by the system. (LOCATE)

STR HAS ONLY n UNITS

You have requested more drives than are necessary to accommodate the structure. The MOUNT proceeds with only the number of drives necessary. (MOUNT)

STRUCTURE ALREADY MOUNTED

[MOUNT COUNT = n]

The requested file structure already exists and does not need to be physically mounted. "n" is the number of users (not including this one) who currently have the structure mounted. (MOUNT)

?STRUCTURE NOT MOUNTED

You requested that a structure be dismounted. The specified structure either is not a valid name or it is not mounted. (DISMOUNT)

?STRUUO FAILURE

The STRUUO UUO gave an error return. Notify the operator. (KJOB, LOGOUT)

%SUPERSEDING EXISTING FILE

A warning message indicating that a file already exists with the specified name. This file is being superseded. (TECO)

%SWAP READ ERROR UNIT abc STATUS = n

An I/O error occurred while reading the swapping space. The data is written into the DAEMON file as read. (DCORE)

?SWITCH ERROR

An illegal switch specification was given. (COPY program)

?switch SWITCH ILLEGAL

The switch specified cannot be used with the given queue name. (QUEUE)

?SWITCH VALUE TOO LARGE x

The value given to the switch exceeds the maximum value. (QUEUE)

?SYNTAX ERROR

There is a syntax error in the command string. Check for incorrect parentheses or two operators in a row.

?SYSSTR FAILURE

The SYSSTR UUO gave an error return. Notify the operator. (KJOB, LOGOUT)

?SYSTEM ERROR - xxxxxx

System errors designate operator or system errors and are not a direct fault of the user. They are typed for possible diagnostic use.

SYSTEM ERROR AT LOC nnn

This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. (FILE, MOUNT, DISMOUNT)

?file structure name SYSTEM ERROR - DSKCHK UUO FAILED

This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. (MOUNT, DISMOUNT)

?file-structure name SYSTEM ERROR - JOBSTR UUO FAILED

This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. (MOUNT, DISMOUNT)

?SYSTEM ERROR - SYSPHY UUO FAILED

This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. (FILE, MOUNT, DISMOUNT)

?SYSTEM ERROR - WRONG FORMAT VERSION NUMBER IN STRLST.SYS

STRLST.SYS is not in a format that is understood by MOUNT. This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. (FILE, MOUNT, DISMOUNT)

?SYSTEM I/O ERROR AT loc STATUS = code

An unexpected I/O error occurred at location "loc". "code" is the status code returned by the GETSTS UUO. This is a system error and therefore is not a direct fault of the user. It is typed for potential diagnostic use. Consult the system administrator. (FILE, MOUNT, DISMOUNT)

?SYSTEM NOT AVAILABLE

The operator has used the SET SCHED command to prevent LOGINs from timesharing terminals. The message of the day is still typed. (LOGIN)

?TABLE OVERFLOW - CORE UUO FAILED TRYING TO EXPAND TO xxx

The GLOB program requested additional core from the monitor, but none was available. (GLOB)

?THIS MONITOR WAS BUILT FOR A xxx AND WILL NOT RUN PROPERLY ON A yyy

The monitor is not running on the machine for which it was built. xxx and yyy are PDP-6, KA10, or KI10.

?TIME LIMIT EXCEEDED

The time limit allocated for the job has been reached. The job is stopped and the terminal is returned to monitor mode.

TIMESHARING WILL CEASE IN m HOURS n MINUTES

The KSYS command (OPSER) or SET KSYS UUO has been issued in order to stop timesharing on the system at the indicated time.

?TOO FEW ARGUMENTS

A command has been typed, but necessary arguments are missing.

?TOO MANY FILENAMES OR PROGRAM NAMES

More than 40 program names or filenames were specified in the command string. The user should separate the job into several segments. (MAKLIB)

?TOO MANY FILES, LAST PROCEED IS file.ext

The user requested (either explicitly or implicitly) more than 22 files to be filed onto or recalled from DECtape. The request is processed up to the point where "file.ext" is transferred. (FILE)

?TOO MANY FILE STRUCTURES

The number of file structures exceeds the capacity of the monitor data base. The current limit is 14 (decimal). (ONCE ONLY)

?TOO MANY NAMES or ?TOO MANY SWITCHES

Command string complexity exceeds table space in the COMPIL program. (COMPIL)

?UNKNOWN COMMAND

The monitor passed a command to COMPIL which COMPIL does not recognize. (COMPIL)

?UNKNOWN DEFAULT FOR SWITCH switch

The default condition is not known for the specified switch. (DUMP, QUEUE)

?UNKNOWN DEVICE OR STRUCTURE NAME

The device or file structure name does not exist in the system administrator's file SYS:STRLST.SYS and, therefore, is not defined for the system. The operator or administrator may be requested to define the file structure by adding it to STRLST.SYS with the REACT program. (MOUNT)

?UNKNOWN SWITCH switch

The switch named has been mistyped. (DUMP, QUEUE)

?UNKNOWN SWITCH VALUE n

The argument specified with the switch has been mistyped. (DUMP, QUEUE)

?UNRECOGNIZABLE SWITCH

An ambiguous or undefined word followed a slash. (COMPIL)

?UUO AT USER adr

This message accompanies many error messages and indicates the location of the UUO that was the last instruction the user program executed before the error occurred.

n VERIFICATION ERRORS

On a word by word comparison requested via the /V switch, n discrepancies have been detected between the input DECtape and output DECtape. (DTCOPY program)

WAITING...2 C's TO EXIT

A request has been queued to the operator and the command is waiting for the operator to complete the request. If you do not want to wait for completion of the operator's action, you can type two CTRL/Cs without aborting the command. The operator action will still be completed. Later a DISMOUNT/CHECK or MOUNT/CHECK can be given to check for completion. (Note that this message is not issued when the /NOWAIT switch is specified.) (MOUNT, DISMOUNT)

WAIT PLS

The system's primary accounting file FACT.SYS was busy. It is retried for ten seconds before FACT.X01 is tried. This message can appear if many users are logging in simultaneously. (LOGIN, KJOB, LOGOUT)

%WARNING - INPUT REQUEST USES ONLY TWO ENTRIES

Only two files can be specified in the input queue request, the control file and the log file. (QUEUE)

!WARNING NO INDEX ON OUTPUT FILE-CONTINUING

You have changed the structure of the index library file when deleting, appending, or inserting, thereby invalidating the index. The index has been removed from the new file. Reindexing is required. (MAKLIB)

?dev WASN'T ASSIGNED

The device is not currently assigned to your job and cannot be deassigned or reassigned by the job.

?WASN'T DET

The specified device is not detached.

```
                                        NAME
?WILDCARD ILLEGAL IN INPUT QUEUE FILE   DIRECTORY
                                        EXTENSION
```

The wildcard construction cannot be used when specifying the Batch input queue. (QUEUE)

```
                               NAME
?WILDCARD ILLEGAL IN OUTPUT    DIRECTORY
                               EXTENSION
```

The wildcard construction cannot be used in the output queue specification. (QUEUE)

%WLDNSF NO SUCH FILES AS file spec

A file has been specified which does not exist in the user's default directory.

?WRITE LOCK ERROR

An attempt was made to write on a write-locked DECtape. (DTCOPY program)

?WRONG FORMAT FOR SYMBOL

A symbol was given in the format program :symbol and a symbol name did not follow the colon; in other words, the colon must be followed by a symbol. (DUMP)

?WRONG FORMAT VERSION NUMBER IN SYSTEM FILES

Wrong version of ACCT.SYS or AUXACC.SYS is on the system. Consult the operator so that he can REACT to change the accounting files. (LOGIN)

drive WRONG UNIT TYPE

The user has requested a drive which is inappropriate for the pack to be mounted. (For example, an RP03 pack cannot be mounted on an RP04 drive.) (MOUNT)

YOU ARE LOGGED IN AS n,m

When you log in with a unique programmer number (project, #), this message informs you of the project-programmer number that LOGIN assigned. (LOGIN)

```
                                  DAEMON
?YOU DON'T HAVE PRIVILEGES TO WRITE        FILE
                                  CCL
```

You attempted to write in a file to which you did not have access. (DAEMON)

?1+1nK CORE
VIR.CORE LEFT = 0

The swapping space or the core allocated to timesharing is all in use (i.e., there is no available virtual core). The user should wait a few minutes, and then attempt to login again. If this message still appears, it should be reported to the operator.

m+n/p CORE
VIR.CORE LEFT = v

Key: m = number of blocks in low segment.

n = number of blocks in high segment.

p = maximum core per job. (Maximum physical user core unless limited by operator, or there are jobs locked in core (refer to DECsystem-10 Monitor Calls.))

v = number of K blocks unassigned in core and on the swapping device.

Note that nK represents 1024-word blocks which is the unit of core allocation on a KA10-based system, and nP represents 512-word blocks which is the unit of allocation on a KI10-based system.

APPENDIX A

STANDARD SYSTEM NAMES

A.1 FILENAME EXTENSIONS

Table A-1 lists the filename extensions that have specific meanings in the DECsystem-10.

Table A-1
Filename Extensions

Filename Extension	Type of File	Meaning
ABS	Object	Absolute (nonrelocatable) program
AID	Source	Source file in AID language
ALG	Source	Source file in ALGOL language
ALP	ASCII	Printer forms alignment
ATO	ASCII	OPSER automatic command file
AWT	Binary	Data for automatic wire tester
B10	Source	Source file in BLIS10
B11	Source	Source file in BLISS-11
BAC	Object	Reserved for output from the BASIC Compiler
BAK	Source	Backup file from TECO or LINED
BAS	Source	Source file in BASIC language
BCM	ASCII	Listing file created by FILCOM (binary compare)
BCP	Source	Source file in BCPL language
BIN	Binary	Binary file
BLB	ASCII	Blurb file
BLI	Source	Source file in BLISS language
BUG	Object	Saved to show a program error
BWR	ASCII	Beware file listing warnings about a file or program
CAL	Object	CAL data and program files
CBL	Source	Source file in COBOL language
CCL	ASCII	Alternate convention for command file (@ command file construction for programs other than COMPIL)
CCO	ASCII	Listing of modifications to nonresident software
CDP	ASCII, Binary	Spooled output for card punch
CFC	ASCII	Compressed file compare. Group of .SCM files combined with PIP.
CKP	Binary	Checkpoint core image file created by COBOL operating system
CHN	Object	CHAIN file

Table A-1 (Cont.)
Filename Extensions

Filename Extension	Type of File	Meaning
CMD	ASCII	Command file for indirect commands (@ construction for COMPIL)
CMP	ASCII	Complaint file by GRIPE
COR	ASCII	Correction file for SOUP
CRF	ASCII	CREF (cross-reference) input file
CTL	ASCII	MP^B control file
DAE	Binary	Default output for DAEMON-taken core dumps
DAT	ASCII, Binary	Data (FORTRAN) file
DDT	ASCII	Input file to FILDDT
DCT	ASCII	Dictionary of words
DIR	ASCII	Directory from FILE command or DIRECT program
DMP	ASCII	COBOL compiler dump file
DOC	ASCII	Listing of modifications to the most recent version of the software
DRW	Binary	Drawing for VB10C drawing system
DSE	ASCII	Directory sorted by extension
DSF	ASCII	Directory sorted by filename
ERR	ASCII	Error message file
EXE	Object	Executable program
F4	Source	Source file in F40 (FORTRAN) language
FAI	Source	Source file in FAIL language
FCL	Source	Source file in FOCAL language
FFS	ASCII	Fast FORTRAN stream
FLO	ASCII	English language flowchart
FOR	Source	Source file in FORTRAN-10 language
FRM	ASCII	Blank form for handwritten records
FTP	Source	FORTRAN test programs
FUD	ASCII	FUDGE2 listing output
GND	ASCII	List of ground pins for automatic wiretap
HGH	Object	Nonsharable high segment of a two-segment program (created by SAVE command)
HLP	ASCII	Help files containing switch explanations, etc.
IDA	ASCII, Binary	COBOL ISAM data file
IDX	ASCII, SIXBIT	Index file of a COBOL ISAM file
INI	ASCII, Binary	Initialization file
LAP	ASCII	Output from the LISP compiler
LIB	ASCII	COBOL source library
LOG	ASCII	MPB or LINK-10 log file
LOW	Object	Low segment of a two-segment program (created by SAVE command)
LPT	ASCII	Spooled output for line printer
LSD	ASCII	Default output for DUMP program
LSP	Source	Source file in LISP language
LSQ	ASCII	Queue listing created by QUEUE program
LST	ASCII	Listing data created by assemblers and compilers
MAC	Source	Source file in MACRO language
MAN	ASCII	Manual (documentation) file
MAP	ASCII	Loader or LINK-10 map file

Table A-1 (Cont.)
Filename Extensions

Filename Extension	Type of File	Meaning
MEM	ASCII	Memorandum file typically output from RUNOFF
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
N	Source	Source file in NELIAC language
NEW	All	New version of a program or file
OBJ	Object	PDP-11 relocatable binary file
OLD	Source, Object	Backup source program
OPR	ASCII	Installation and assembly instructions
OVR	Object	COBOL overlay file
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)
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
RNP	ASCII	RUNOFF input for producing a .OPR file
RNS	ASCII	RUNOFF input for text file of standards
RSP	ASCII	Script response time log file
RSX	All	Files for RSX-11D
RTB	Object	Read-in mode (RIM10B) format file (PIP)
SAI	Source	Source file in SAIL language

Table A-1 (Cont.)
Filename Extensions

Filename Extension	Type of File	Meaning
SAV	Object	Low segment from a one-segment program (created by SAVE command)
SCD	ASCII	Differences in directory
SCM	ASCII	Listing file created by FILCOM (source compare)
SCP	ASCII	SCRIPT control file
SEQ	ASCII, SIXBIT	Sequential COBOL data file, input to ISAM program
SFD	Binary	Subfile directory (reserved usage)
SHR	Object	Sharable high segment file of a two-segment program (created by SSAVE command)
SMP	Source	Source file in SIMPLE language
SNO	Source	Source file in SNOBOL language
SNP	ASCII	Snapshot of disk by DSKLST
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
TMP	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
UFD	Binary	User file directory (reserved usage)
UPD	ASCII	Updates flagged in margin (FILCOM)

Table A-1 (Cont.)
Filename Extensions

Filename Extension	Type of File	Meaning
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)

A.2 RESERVED PROJECT-PROGRAMMER NUMBERS

Table A-2 itemizes the project-programmer numbers that are allocated for specific functions in the DECsystem-10.

Table A-2
Project-Programmer Numbers

Number	Meaning
1,1	Master File Directory (MFD)
1,2	Operator functions
1,3	Old or superseded versions of system programs (device OLD:)
1,4	System library (device SYS:)
1,5	New or experimental versions of system programs (device NEW:)
1,6	User maintained library (device PUB:)
2,*	Recommended for operator's use
2,5	Storage for help text files (*.HLP) (device HLP:)
3,3	System and Multiprogram Batch (MPB) queues
4,*	Test and performance analysis systems
4,4	FAILSAFE testing
4,5	FAILSAFE testing
5,*	Libraries
5,1	BASIC source library (device BAS:)
5,2	COBOL source library for COPY verb (device COB:)
5,3	PDP-11 source library (device MXI:)
5,4	ALGOL source library (device ALG:)
5,5	BLISS source library (device BLI:)
5,6	FORTRAN source library (device FOR:)
5,7	MACRO source library (device MAC:)
5.10	Text editor library (device TED:)
5,11	Rel file library (device REL:)
5,12	RUNOFF library (device RNO:)
5,13	SNOBOL library (device SNO:)
5,14	Doc file library (device DOC:)
5,15	FAIL library (device FAI:)

Table A-2 (Cont.)
Project-Programmer Numbers

Number	Meaning
5,16	MUSIC library (device MUS:)
5,17	MACRO universal files (device UNV:)
5,20	NELIAC (device NEL:)
5,21	DUMP (device DMP:)
5,22	POP2 (device POP:)
5,23	Test library (device TST:)
6,*	Field service and hardware diagnostics
7,7	Software acceptance
10,1	Special system programming storage region containing copies of SYS:CRASH.SAV (device XPN:)
10,6	Software distribution
10,7	Software distribution (device DEC:)

A.3 RESERVED DEVICE NAMES

Table A-3 lists logical device names (ersatz devices) that are predefined in the DECsystem-10.

To avoid confusion, it is recommended that these names not be used for private file structures.

Table A-3
Device Names

Name	Use	UFD	Search List
ALG:	ALGOL library	[5,4]	System
ALL:		User's	All currently mounted structures
BAS:	BASIC library	[5,1]	System
BLI:	BLISS library	[5,5]	System
COB:	COBOL library	[5,2]	System
DEC:	DEC supplied software	[10,7]	System
DOC:	DOC file library	[5,14]	System
DSK:		User's	Job
FAI:	FAIL library	[5,15]	System
FOR:	FORTRAN library	[5,6]	System
HLP:	HELP library	[2,5]	System
LIB:	User defined library	Set by each user	System
MAC:	MACRO library	[5,7]	System
MUS:	MUSIC library	[5,16]	System
MXI:	PDP-11 library	[5,3]	System
NEW:	New system library	[1,5]	System
NUL:		None	
OLD:	Old system library	[1,3]	System
PUB:	User maintained library	[1,6]	System
REL:	Rel file library	[5,11]	System
RNO:	RUNOFF library	[5,12]	System

Table A-3 (Cont.)
Device Names

Name	Use	UFD	Search List
SNO:	SNOBOL library	[5,13]	System
STD:	Standard Software	[1,4]	System
SYS:	System library	[1,4]	System
TED:	Text editor library	[5,10]	System
TST:	Test library	[5,23]	System
UNV:	MACRO universal library	[5,17]	System
XPN:	Crash library	[10,1]	System

APPENDIX B

CARD CODES

Table B-1
ASCII Card Codes

ASCII Character	Octal Code	Card Punches	ASCII Character	Octal Code	Card Punches
NULL	00	12-0-9-8-1	@	100	8-4
CTRL-A	01	12-9-1	A	101	12-1
CTRL-B	02	12-9-2	B	102	12-2
CTRL-C	03	12-9-3	C	103	12-3
CTRL-D	04	9-7	D	104	12-4
CTRL-E	05	0-9-8-5	E	105	12-5
CTRL-F	06	0-9-8-6	F	106	12-6
CTRL-G	07	0-9-8-7	G	107	12-7
CTRL-H	10	11-9-6	H	110	12-8
TAB	11	12-9-5	I	111	12-9
LF	12	0-9-5	J	112	11-1
VT	13	12-9-8-3	K	113	11-2
FF	14	12-9-8-4	L	114	11-3
CR	15	12-9-8-5	M	115	11-4
CTRL-N	16	12-9-8-6	N	116	11-5
CTRL-O	17	12-9-8-7	O	117	11-6
CTRL-P	20	12-11-9-8-1	P	120	11-7
CTRL-Q	21	11-9-1	Q	121	11-8
CTRL-R	22	11-9-2	R	122	11-9
CTRL-S	23	11-9-3	S	123	0-2
CTRL-T	24	9-8-4	T	124	0-3
CTRL-U	25	9-8-5	U	125	0-4
CTRL-V	26	9-2	V	126	0-5
CTRL-W	27	0-9-6	W	127	0-6
CTRL-X	30	11-9-8	X	130	0-7
CTRL-Y	31	11-9-8-1	Y	131	0-8
CTRL-Z	32	9-8-7	Z	132	0-9
ESCAPE	33	0-9-7	[	133	12-8-2
CTRL-\	34	11-9-8-4	\	134	0-8-2
CTRL-]	35	11-9-8-5	]	135	11-8-2
CTRL-^	36	11-9-8-6	^	136	11-8-7
CTRL-	37	11-9-8-7	_	137	0-8-5
SPACE	40		`	140	8-1

NOTE

The ASCII character ESCAPE (octal 33) is also CTRL-[on a terminal.

Table B-1 (Cont.)
ASCII Card Codes

ASCII Character	Octal Code	Card Punches	ASCII Character	Octal Code	Card Punches
!	41	12-8-7	a	141	12-0-1
'	42	8-7	b	142	12-0-2
#	43	8-3	c	143	12-0-3
$	44	11-8-3	d	144	12-0-4
%	45	0-8-4	e	145	12-0-5
&	46	12	f	146	12-0-6
'	47	8-5	g	147	12-0-7
(	50	12-8-5	h	150	12-0-8
)	51	11-8-5	i	151	12-0-9
*	52	11-8-4	j	152	12-11-1
+	53	12-8-6	k	153	12-11-2
,	54	0-8-3	l	154	12-11-3
-	55	11	m	155	12-11-4
.	56	12-8-3	n	156	12-11-5
/	57	0-1	o	157	12-11-6
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.

CARD CODES

Table B-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 0-37 and 140-177 are the same as in ASCII.

APPENDIX C

TEMPORARY FILES

The temporary files in Table C-1 are used by various programs in the DECsystem-10 computing system. These files are in the following form:

```
nnn xxx.TMP
```

where nnn is the user's job number in decimal, with leading zeroes to make three digits, and xxx specifies the use of the file.

Table C-1
Temporary Files

Name	Meaning
nnn ALG.TMP	Read by ALGOL and contains one line for each program to be compiled. It may also contain the command NAME! which causes ALGOL to transfer control to the named program.
nnn AS1.TMP nnn AS2.TMP nnn AS3.TMP	Written, read, and deleted by COBOL and contains input to the COBOL assembler.
nnn BL1.TMP	Read by BLISS and contains one line for each program to be compiled.
nnn COB.TMP	Read by COBOL and contains one line for each program to be compiled. It may also contain the command NAME! which causes COBOL to transfer control to the named program.
nnn CPY.TMP	Written, read, and deleted by COBOL and contains copies of source files with library routines inserted.
nnn CRE.TMP	Read by CREF and contains commands for each file which has produced a CREF listing on the disk. COMPIL also reads this file each time a new CREF listing is generated to prevent multiple requests for the same file and to prevent discarding other requests that may not yet have been listed.
nnn DAE.TMP	Written by DAEMON to be read by DUMP.
nnn DMP.TMP	Read by DUMP as an input command file.
nnn EDS.TMP	Used by COMPIL to store the arguments of the most recent EDIT, CREATE, TECO, or MAKE command.

Table C-1 (Cont.)
Temporary Files

Name	Meaning
nnn EDT.TMP	Written by COMPIL and read by LINED or TECO. It contains a command for each EDIT, CREATE, TECO, or MAKE command. For the MAKE or CREATE commands, it contains the command S file.ext [p,p] $ For TECO or EDIT commands, it contains the command S file.ext [p,p]
nnn ERA.TMP	Written, read, and deleted by COBOL and is the error file.
nnn FA1.TMP	Read by FAIL and contains one line for each program to be compiled.
nnn FOR.TMP	Read by FORTRAN and contains one line for each program to be compiled. It may also contain the command NAME! which causes FORTRAN to transfer control to the named program.
nnn GEN.TMP	Written, read, and deleted by COBOL and contains the output of syntax processing.
nnn KJO.TMP	Read by KJOB as an input command file.
nnn LGO.TMP	Read by LOGOUT as an input command file.
nnn LHC.TMP	Created and read by LINK-10 and contains the overflow of the user's high segment. The file is used to produce core images or saved files.
nnn LIN.TMP	Created by LINED and contains output file until the rename process.
nnn LIT.TMP	Written, read, and deleted by COBOL and contains copy of the literal pool.
nnn LLC.TMP	Created and read by LINK-10 and contains the overflow of the user's low segment. This file is used to produce core images or saved files.
nnn LLS.TMP	Created and read by LINK-10 and contains the overflow of the user's symbol file. This file is used to produce core images or saved files.
nnn LNK.TMP	Read by LINK-10 and contains commands necessary for loading.
nnn LOA.TMP	Read by LOADER and contains commands necessary for loading.
nnn MAC.TMP	Read by MACRO and contains one line for each program to be assembled. It may also contain the command NAME! which causes MACRO to transfer control to the named program.
nnn P11.TMP	Read by MACY11 (the PDP-11 assembler for the PDP-10) and contains one line for each program to be assembled.
nnn PLS.TMP	Read by PLEASE as an input command file.
nnn PIP.TMP	Read by PIP and contains commands to implement the COMPIL-class commands that run PIP.
nnn QUE.TMP	Read by QUEUE as an input command file.
nnn RNO.TMP	Read by RUNOFF and contains commands for each file which has produced a RUNOFF listing on the disk.
nnn S01.TMP	Written, read and deleted by COBOL and contains the intermediate sorted results of the data.

Table C-1 (Cont.)
Temporary Files

Name	Meaning
nnn SVC.TMP	Used by COMPIL to store the arguments of the most recent COMPIL, LOAD, EXECUTE, or DEBUG command.
nnn SNO.TMP	Read by SNOBOL and contains one line for each program to be compiled.
nnn TEC.TMP	Created by TECO and contains output file until the rename process.
nnn TMP.TMP	Created by LINED during the rename process.
nnn XFO.TMP	Created by FILEX as a result of the Q switch on the output side.
nnn XFR.TMP	Created by FILEX as a result of the Q switch on the input side.

APPENDIX D

SIXBIT/ASCII CHARACTER CODES

SIXBIT	Character	ASCII 7-Bit	SIXBIT	Character	ASCII 7-Bit	Character	ASCII 7-Bit
00	Space	040	40		100	´	140
01	!	041	41	A	101	a	141
02	"	042	42	B	102	b	142
03	#	043	43	C	103	c	143
04	$	044	44	D	104	d	144
05	%	045	45	E	105	e	145
06	&	046	46	F	106	f	146
07	'	047	47	G	107	g	147
10	(	050	50	H	110	h	150
11	)	051	51	I	111	i	151
12	*	052	52	J	112	j	152
13	+	053	53	K	113	k	153
14	,	054	54	L	114	l	154
15	-	055	55	M	115	m	155
16	.	056	56	N	116	n	156
17	/	057	57	O	117	o	157
20	0	060	60	P	120	p	160
21	1	061	61	Q	121	q	161
22	2	062	62	R	122	r	162
23	3	063	63	S	123	s	163
24	4	064	64	T	124	t	164
25	5	065	65	U	125	u	165
26	6	066	66	V	126	v	166
27	7	067	67	W	127	w	167
30	8	070	70	X	130	x	170
31	9	071	71	Y	131	y	171
32	:	072	72	Z	132	z	172
33	;	073	73	[	133	{	173
34	<	074	74	\	134	\	174
35	=	075	75	]	135	}	175
36	>	076	76	↑	136	∿	176
37	?	077	77	←	137	Delete	177

INDEX

DECsystem-10
Operating System
Commands Manual
AA-0916C-TB

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

Did you find errors in this manual? If so, specify by page.

____
____
____
____
____
____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____
____
____
____
____
____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

____
____
____
____
____
____

Please indicate the type of user/reader that you most nearly represent.

□ Assembly language programmer
□ Higher-level language programmer
□ Occasional programmer (experienced)
□ User with little programming experience
□ Student programmer
□ Non-programmer interested in computer concepts and capabilities

Name ______________________ Date ______________

Organization ______________________________

Street ______________________________

City ______________ State ____________ Zip Code or Country ______________

Please cut along this line.

-- Fold Here --

-- Do Not Tear - Fold Here and Staple --

FIRST CLASS
PERMIT NO. 152
MARLBORO, MASS

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
200 Forest Avenue MR1-2/E37
Marlboro, Massachusetts 01752