digital

software

OS/78
User's Manual

Order No. AA-5748B-TA

August 1979

This document is the user's manual for the OS/78 V3 operating system. Its purpose is to acquaint new users with the operating system designed for the DECstation 78/88 minicomputers. This manual presents the background material for getting on the air, and a detailed description of the OS/78 commands that are used to direct computer operations. It also describes the OS/78 Editor, PAL8 assembly language, and the two high-level languages supported by the system, BASIC and FORTRAN IV. Command examples and demonstration programs are contained throughout the manual.

SUPERSESSION/UPDATE INFORMATION: This manual supersedes previous editions, Order Numbers DEC-S8-OS78A-A-D, published 1977, and DEC-S8-OS78A-A-DN1, published 1978.

OPERATING SYSTEM AND VERSION: OS/78 V3

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

digital equipment corporation · maynard. massachusetts

First Printing, August 1979

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

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	

PREFACE

This manual describes the OS/78 Operating System for the DECstation 78 and DECstation 88 Minicomputer systems.

SYNOPSIS

Chapter 1 briefly describes the system hardware configurations and lists the software contained on the master disk distribution media.

Chapter 2 contains the basic instructions that describe how to use the OS/78 Operating System. You should read this chapter in its entirety to gain familiarity with the operating characteristics and features of the system prior to using OS/78.

Chapter 3 describes in alphabetic order the commands that direct computer operations.

Chapter 4 describes the OS/78 Editor, a program that lets you create and modify ASCII source files.

Chapter 5 describes PAL8, a symbolic assembler.

Chapters 6 and 7 describe the system's two high level programming languages, BASIC and FORTRAN IV, that run under OS/78.

Chapter 8 describes BATCH, the system's batch processing program.

Chapter 9 contains information on how to use ODT (Octal Debugging Technique) program to change and correct memory-resident PAL8 programs.

The appendixes contain information for experienced users about device handlers, the User Service Routine, the Command Decoder, useful mathematical subroutines, an alphabetic summary of all OS/78 error messages, and a description of the DECstation 78 Symbiont operation.

NOTE

> For historic reasons, the words "core" and "core memory" appear in this manual. You should interpret these terms as meaning any kind of memory. There is no implication that your DECstation uses core memory.

RELATED HANDBOOKS AND MANUALS

DECscope User's Manual (VT-52) (EK-VT5X-OP-001)
DECstation/78 User's Guide (EK-VTX-78-TM-OP-001)
Introduction to Programming (DEC-08-XINPA-A-D)
KT8A Memory Management Control User's Guide (EK-KT08A-UG-001)
LA34/38 User's Guide (EK-0LA34-UG-001)
LA36 User's Guide (EK-LA365-OP-002)
LA120 User's Guide (EK-LA120-UG-001)
LA180 User's Guide (EK-LA180-OP-002)
LQP8 Printer Systems Technical Manual (EK-LQP8-TM-001)
OS/8 Error Messages Manual (AA-H610A-TA)
OS/8 System Generation Notes (AA-H606A-TA)
OS/8 System Reference Manual (AA-H607A-TA)
RX02 Floppy Disk System User's Guide (EK-RX02-UG-001)
RX8/RX11 Floppy Disk System User's Manual (EK-RX01-OP-001)
VT100 User's Guide (EK-VT100-UG-001)

DOCUMENT CONVENTIONS

This section describes the conventions and symbology used in this manual.

- The portions of the command keywords that appear in red identify the parts of a keyword that you must use. For example:

  ```
  DIRECT dev:/options
  ```

  The portions of the command keywords that appear in black are optional.

  The portions of the examples that appear in red indicate the parts of an example that you must type into the system from the keyboard. The portions of the examples that appear in black identify the text that the system and its programs display. For example:

  ```
  .DATE
  NONE
  .
  ```

In this example, the monitor displays the period, its prompting symbol. You type DATE (and then press the RETURN key). The system displays NONE; the monitor again displays the period (.).

- In the command formats, upper case characters indicate a keyword that you must enter exactly as shown. Lower-case characters indicate variable information that you must supply. For example:

  ```
  DUPLICATE outdev:<indev:/options
  ```

- Double square brackets ([[ ]]) enclose optional items.

- Ellipses (...) indicate that you may repeat the preceding item or bracketed group any number of times.

For example in a FORTRAN program, if the description is

CALL sub [[(a [[,a [[...)]]

then all of the following are correct:

```
CALL TIMER
CALL INSPCT (I,J,3.0)
CALL REGRES (A)
```

The following keyboard characters are designated by symbols to indicate specific operations that you perform when using OS/78.

Key	Symbol	Function
DELETE	(DEL)	Deletes a character
ESC	(ESC)	Performs special functions when using PAL8, FORTRAN and the OS/78 Editor
LINE FEED	(LF)	Performs a line feed
RETURN	(RET)	Performs a carriage or cursor return to the left margin. In this manual, you must follow all command and input lines by a carriage return (pressing the RETURN key) unless otherwise specified by the appearance of one of the symbols shown above.
TAB	(TAB)	Skips up to eight spaces

CHAPTER 1

INTRODUCTION TO THE SYSTEM

The OS/78 operating system is a software product designed for the DECstation 78 and DECstation 88 Minicomputer systems. OS/78 permits you to use a variety of peripheral devices including: video and hard copy terminals, cartridge and flexible media disk drives, line and letter quality printers, and up to 32K words of memory.

OS/78 offers a versatile set of keyboard commands that supervise a comprehensive library of system programs. These programs allow you to develop your application programs using the PAL8 assembly language, or the higher level BASIC, and FORTRAN IV languages.

The system also has a batch stream processor, called BATCH, that provides for unattended (batch-mode) system operation.

1.1 DECSTATION 78/88 MINICOMPUTER SYSTEM HARDWARE

The DECstation 78/88 Minicomputer systems consist of four basic units:

- a video or hard copy console terminal
- a disk system (either flexible or cartridge media)
- a central processor
- main memory

In addition, the system can support a line printer (LA78, LA180, or LQP78), additional disk drives, and auxiliary terminals connected to the serial asynchronous interface ports. These ports are suitable for primary and secondary communications applications, terminals, and a variety of user-provided devices.

The console terminal provides the means for interacting with or "talking" to the computer. It serves as the primary source of OS/78 command inputs. There are three types of terminals: the VT78 and VT100 video terminals, and the LA36 hard-copy printer.

The disk system can consist of one or two dual disk drives that use flexible (floppy) media (RX01 and RX02), one or two disk drives that use removable cartridge media (RL01), or a combination of both types. The RX01 floppy disk drive uses a diskette formatted for single-density operation. The RX02 floppy disk drive uses a diskette formatted for either double-density or single-density operation.

The main functions of the disk system are:

- To contain a prerecorded copy of the OS/78 software that runs the computer system.
- To provide space to store the programs and files that you create.

The central processing unit (CPU) executes instructions from main memory and controls all peripheral devices. On DECstation 78 systems, the CPU is built into the VT78 terminal. On DECstation 88 systems it resides in a cabinet with the disk system.

The main memory is the active storage area in the computer from which the CPU fetches and executes instructions and into which your programs store and manipulate data.

Appendix I summarizes the DECstation hardware configurations.

For more information on the DECstation Minicomputer System, refer to either the DECstation 78 User's Guide or the DECstation 88 User's Guide as applicable.

1.2 OS/78 SOFTWARE

The OS/78 software can run on DECsystem 78/88 configurations or on any equivalent PDP-8 configuration. The system allows you to write programs in three languages: PAL8 assembly language, BASIC, and FORTRAN IV. You manipulate program source and data files by entering OS/78 commands from the console terminal. The software to do this resides on the master disk media provided with your system.

DIGITAL supplies the OS/78 system software to you on one of three types of disk media. Which medium you get depends on your disk system. If your DECstation has an RX01 floppy disk drive, you will receive the software on two single-density diskettes. If your DECstation has an RX02 floppy disk drive, you will receive the software on one double-density floppy diskette. If your DECstation has an RL01 cartridge disk pack drive only, you will receive the software on one cartridge disk pack.

CHAPTER 2

GETTING ON THE AIR WITH OS/78

This chapter provides the basic information that you need to use the OS/78 operating system. You should read the entire chapter prior to using the system to become familiar with the features and capabilities of OS/78.

2.1 STARTING THE SYSTEM

This section tells you how to load and start your system and to make backup copies of the master disk media. Before proceeding, be sure that your DECstation is connected to an appropriate alternating current power source and the power switch is in the ON position (refer to the applicable hardware manual listed in the Preface).

2.1.1 Loading the Diskette

Since the diskette is thin and flexible, be careful when handling and storing it. Hold it by the edges and do not touch the exposed portions that show through the protective cover (see Figure 2-1). Also, use only felt-tip pens when writing on diskette labels. Store them in a dust-free area, and do not expose them to magnetic devices, direct sunlight, heat, or humidity. Refer to the RX8 and RX28 User's Guides (see Preface) for further information on the care of diskettes.

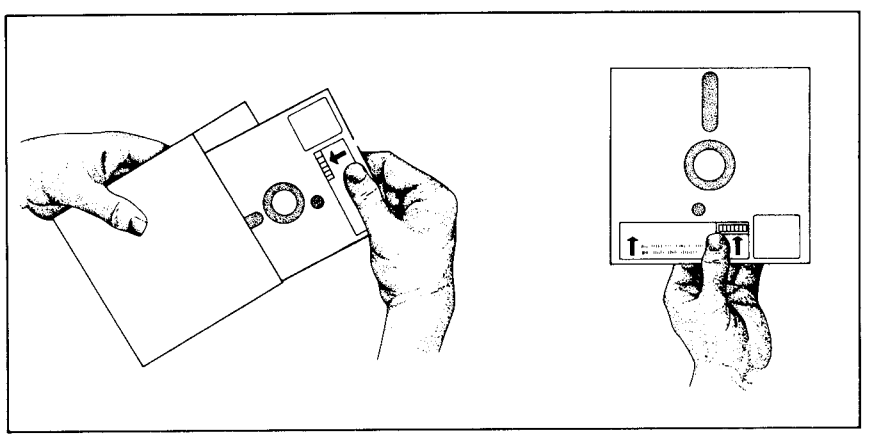


Figure 2-1 Handling the Diskette

NOTE

Do not insert a double-density diskette into a single-density drive (RX01). The single-density drive cannot read a double-density diskette.

The following steps and Figure 2-2 describe how to load a diskette into an RX01 or RX02 disk drive:

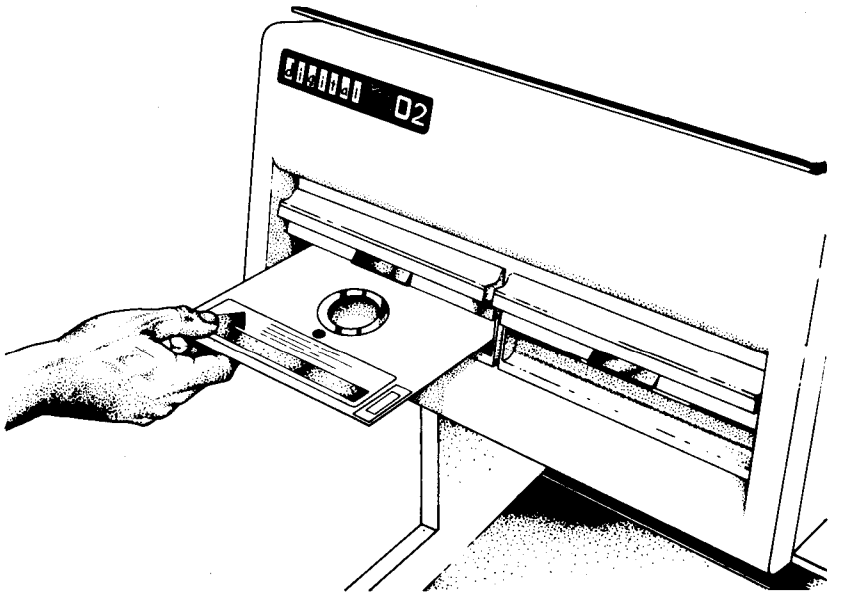


Figure 2-2 Loading the Diskette

1. Open the door:

   a. Notice that the door latch is embedded in the horizontal bar that is part of the door assembly.

   b. Compress the latch between the thumb and index finger.

   c. The door is spring-loaded and will open when the latch disengages.

2. Remove the diskette from its storage jacket. (This is not the black protective cover in which the diskette is permanently sealed.)

3. Insert the diskette label side up (the jacket seams are on the bottom) into the horizontal slot in the drive.

4. Close the door. It will automatically lock when you push it down.

Ordinarily you should load the diskette that contains your system programs (system diskette) into Drive 0, which is the left-hand slot. You can use Drive 1 (right hand slot) if necessary, so long as there is no diskette in Drive 0 when you press the START button.

2.1.2 Loading the RL01K Cartridge Disk Pack

The following steps and Figure 2-3 describe how to load the RL01K cartridge disk pack into its drive. If you are loading the disk pack that contains the OS/78 system programs (system disk), it must reside on RL01 Drive 0. The drive number is embossed on the READY indicator lamp next to the LOAD button.

1. If the drive is not the top drive in the cabinet, you must first slide the drive out from the cabinet by grasping it by the horizontal slot in the front panel and gently pulling it toward you.

   Release the drive's cover by sliding the latch on the right side of the cover away from you with your thumb; then raise the cover. If the slide will not operate, the drive currently has a disk pack loaded in it and is on line. Follow the procedures in Section 2.1.3 to unload the pack from the drive.

2. Separate the protective cover from the disk pack as follows:

   a. Lift the cartridge by grasping the handle with your right hand.

   b. Support the cartridge from underneath with your left hand.

   c. Lower the handle, then push the handle slide to its extreme position with your right thumb while raising the handle to its upright position.

   d. Lift the cartridge from the protective cover.

3. Place the cartridge into the drive shroud with the handle recess facing the rear of the machine.

4. Rotate the cartridge a few degrees clockwise and then counter-clockwise to ensure that it is properly seated within the shroud.

5. Gently lower the handle to a horizontal position to engage the drive spindle.

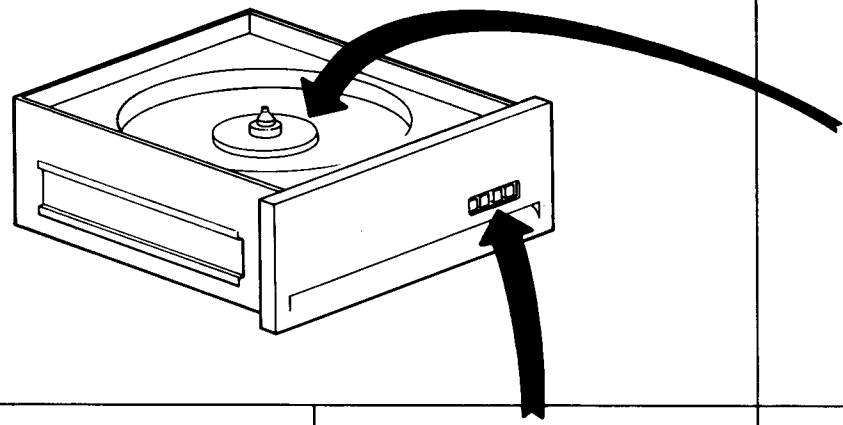
6. Place the protective cover on top of the cartridge.

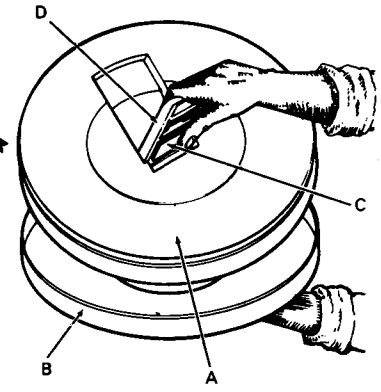
7. Lower the drive's cover carefully and be sure that the latch engages with the case.

8. Press the LOAD button so that it engages in its innermost position. (The LOAD lamp will extinguish.) When the drive is up to speed and on line, the READY lamp will light.

CAUTION

The OS/78 system will not operate properly if you enable the write protect feature of RL01 Drive 0 (WRITE PROT button illuminated).





TO READY DRIVE:

- RAISE CARTRIDGE ACCESS DOOR
- LOAD RL01K DISK CARTRIDGE
- DEPRESS RUN/STOP SWITCH (LOAD INDICATOR)
- NOTE THAT SPINDLE MOTOR STARTS TURNING
- AFTER 30 SECONDS, UNIT SELECT INDICATOR SHOULD LIGHT INDICATING DRIVE IS READY TO READ OR WRITE
- IF WRITE PROTECTION IS DESIRED, DEPRESS WRITE PROTECT SWITCH (PROTECT INDICATOR)

DRIVE INDICATORS:

LOAD:
LIGHTS TO INDICATE THAT CARTRIDGE MAY BE LOADED OR THAT SPINDLE IS STOPPED.

UNIT SELECT:
INDICATES LOGICAL DRIVE ADDRESS. WHEN LIT, INDICATES DRIVE IS READY TO READ, WRITE OR RECEIVE CONTROLLER COMMANDS.

FAULT:
WHEN LIT, INDICATES A DRIVE ERROR CONDITION. IF THIS CONDITION PERSISTS, SEEK ASSISTANCE.

WRITE PROTECT:
WHEN LIT, INDICATES THAT CARTRIDGE CURRENTLY MOUNTED IS WRITE PROTECTED.

TO LOAD RL01K CARTRIDGE:

- SUPPORT CARTRIDGE "A" WITH LEFT HAND HOLDING PROTECTION COVER "B".
- PUSH HANDLE SLIDE "C" TO LEFT WITH THUMB OF RIGHT HAND
- RAISE COVER HANDLE "D" TO FULL UPRIGHT POSITION, RELEASING PROTECTION COVER "B".
- LIFT CARTRIDGE "A" FROM PROTECTION COVER "B" AND CAREFULLY SEAT IT ON DRIVE SPINDLE WITH HANDLE RECESS FACING REAR OF DRIVE.
- CAREFULLY ROTATE TOP COVER HANDLE "D" A FEW DEGREES CLOCKWISE AND COUNTER-CLOCKWISE TO ENSURE FIRM SEATING.
- GENTLY LOWER TOP COVER HANDLE "D" TO HORIZONTAL POSITION TO ENGAGE CARTRIDGE ON DRIVE SPINDLE.
- PLACE PROTECTION COVER "B" ON TOP OF CARTRIDGE.

Figure 2-3 Loading the RL01K Disk Pack

2.1.3 Unloading the RL01K Cartridge Disk Pack

The following steps describe how to unload the RL01K cartridge disk pack from its drive.

1. If the drive is not the top drive in the cabinet, you must first slide the drive out from the cabinet by grasping it by the horizontal slot in the front panel and gently pulling it toward you.

   Place the unit off line by pressing the LOAD button so that it releases to its outermost position. The READY indicator lamp will extinguish and in a few seconds the LOAD button will light. This indicates that the disk has stopped and you can open the cover.

2. Open the drive's cover by squeezing the latch on the right side of the cover between your thumb and index fingers; then raise the cover.

   If the drive is not the top drive in the cabinet, you must first slide the drive out from the cabinet by grasping the sides of the front panel and gently pulling it toward you.

3. Remove the disk's protective cover from the drive.

4. Disengage the disk pack from the drive spindle as follows:

   a. Push the handle slide to its extreme position with your right thumb while raising the handle to a vertical position.

   b. Lift the cartridge from the drive shroud.

5. Place the disk pack in its protective cover as follows:

   a. Hold the cartridge by its handle in your right hand.

   b. Support the protective cover underneath with your left hand.

   c. Lower the cartridge into the protective cover until you hear the handle slide engage.

2.1.4 Starting the System

2.1.4.1 **Setting Terminal Characteristics** - Before using your terminal, read the instruction manual supplied with it. Each terminal has features that you can control either by setting switches or by typing special setup mode commands. Except for the VT100, you can use your terminal as supplied by DIGITAL once you have set its transmit and receive speeds. Table 2-1 lists the terminal operating parameters necessary for OS/78 operation.

Table 2-1
Terminal Operating Parameters

Terminal Type	Parameters
LA34/38	Transmit Speed = 300 Baud Receive Speed = 300 Baud
LA36	Transmit Speed = 300 Baud Receive Speed = 300 Baud
LA120	Transmit Speed = 1200 Baud Receive Speed = 1200 Baud
VT52	Transmit Speed = 4800 Baud Receive Speed = 4800 Baud
VT100	Transmit Speed = 4800 Baud Receive Speed = 4800 Baud SETUP A Set tab stops as required. SET-UP B Set operating parameters as follows: 1 2 3 4 0x0x xx0x xxxx x01x x= user convenience option

2.1.4.2 **Bootstrapping OS78** - Once you have successfully loaded a system diskette or disk pack into Drive 0, you can start the system by calling the OS/78 Monitor. The monitor is the main control program that you interact with when you use OS/78.

Call the monitor by pressing the START or BOOT button on your DECstation. If you have a DECstation 78 series system, the START button is located on the right-hand side of the VT78 console terminal. If you have a DECstation 88 series system, the BOOT button is located on the front of the cabinet containing the disk(s) to the right of the DECstation 88 logo. After a brief pause the monitor will display the OS/78 command summary on the terminal and then a period (.). The period is a prompting symbol that means that the monitor is waiting for you to enter a command. You can now type any OS/78 command.

When OS/78 starts up, it displays a summary of the commands. This feature is controlled by the SET SYS INIT command (see Chapter 3). If your console terminal is a printer rather than a scope, type the following commands:

```
.SET TTY: NO SCOPE
.SET TTY: WIDTH 132
```

You may also wish to inhibit the command summary to save time and paper. Use the following command:

```
.SET SYS: NO INIT
```

2.1.5 Using the Terminal

The keyboard on the terminal allows you to enter data and OS/78 commands into the computer. Most of the keyboard is identical to a typewriter keyboard. Unlike a typewriter, pressing a key does not automatically display a character on the keyboard screen. Pressing a key sends a character selected to be sent to the central processing unit (CPU). The OS/78 program that is running would normally respond by transmitting this character back to the terminal, thereby displaying it. Consequently, typing a character will have no effect if OS/78 is not running or if OS/78 is busy running a program that is not waiting for input from the keyboard.

Certain keys have special functions when used in OS/78 commands. The keys and their functions are as follows:

RETURN key

Use this key to terminate OS/78 command lines, Editor command lines, program statements, and responses to program input queries.

DELETE key

Use this key to correct typing errors in text that is not yet terminated by a RETURN. Each time you press DELETE, the system deletes one character to the left until it reaches the beginning of the line.

LINE FEED key

Use this key while entering monitor command lines. It causes the monitor to display the command line up to the previous carriage return.

CTRL key

Use this key in combination with another key to enter a special control command. Hold the CTRL key down while striking the other key. In this manual, these combinations are shown with a slash (/) between CTRL and the other key, for example: CTRL/U, CTRL/C (do not type the slash).

CTRL/C Terminates execution of a currently active program, and returns control to the monitor. It also returns control from a BASIC program to the BASIC editor.

CTRL/O Stops the display of characters on the screen.

CTRL/S Suspends terminal output of a program or monitor command but does not terminate the program or command.

CTRL/Q Resumes the terminal output that was suspended by CTRL/S.

CTRL/U Deletes a line typed to the monitor or the Editor.

The CTRL key commands require no terminator; the system performs the function as soon as you type the command.

ESCape key

This key performs special functions when using the Editor and running FORTRAN IV and PAL8 programs. The function of this key is described in those sections where its use is applicable.

2.2 MAKING A BACKUP COPY OF THE OS/78 SOFTWARE

It is important that you make at least one copy of the master disk media supplied with your DECstation so that a copy of the system will be available in case the media that contains your working copy fails. Use the master disk media only to make working copies of the system. Never use it as a working copy or alter it in any way!

The following sections describe the procedures to follow for each DECstation type.

2.2.1 DECstations With RX01 and RX02 Disk Systems Only

Perform the following steps to make a backup copy of your master diskette(s):

1. Place the master diskette in Drive 0, the left-hand slot, and a blank diskette in Drive 1.

2. Start the system by pressing the START button.

3. In response to the monitor's period (.), type the DUPLICATE command followed by a carriage return (pressing the RETURN key):

   ```
   .DUP RXA1:<RXA0:
   ```

   The system will duplicate the contents of the diskette in Drive 0 onto the diskette in Drive 1. When the operation is complete, control returns to the monitor, as indicated by the period (.) prompting symbol.

4. Remove both diskettes, and label the diskette taken from Drive 1 appropriately.

5. Repeat Steps 1 through 4 for master diskette 2 (RX01 systems) or to make additional backup copies.

6. Store the master diskettes in a safe place.

7. Place a copy of the system diskette in Drive 0 for use of the working system diskette.

8. Press the system's START button, and you are ready to use OS/78.

2.2.2 DECstations With RL01 Disk Systems Only

Perform the following steps to make a backup copy of your master disk pack:

1. Place the master disk pack in Drive 0 and a blank disk pack in Drive 1.

2. Start the system by pressing the BOOT button.

3. Type the following commands in response to the monitor's period (.) prompting symbol:

```
.FORMAT RL01
.ZERO RL0A:/Y
.ZERO RL0B:
.ZERO RL0C:
.COPY RL1A:<RL0A:*.*
```

   The system will format the disk pack, zero each device and copy all the system files from unit A of Drive 0 to unit A of Drive 1. When the operation is complete, the monitor takes control as indicated by the period (.).

4. Remove both disk packs and label the disk pack taken from Drive 1 appropriately. This is the working system disk pack.

5. Store the master disk pack in a safe place.

6. Place the working system disk pack in Drive 0.

7. Press the system's BOOT button, and you are ready to use OS/78.

2.2.3 DECstations With RX02 and RL01 Disk Systems

The following steps describe how to make a backup copy of your master disk pack.

1. Place the master disk pack in RL01 Drive 0. Make sure that the drive is not write protected (WRITE PROT indicator lamp is extinguished).

2. Press the BOOT button.

3. Place a blank diskette in RX02 Drive 0.

4. Type the following command to initiate the backup operation:

```
.SUBMIT BUILDX
```

5. When the monitor again displays the period (.), remove the disk pack from its drive, and place a blank disk pack in the drive.

6. Type the following command to complete the backup operation:

```
.SUBMIT BACKRL
```

7. When the monitor again displays the period (.), the disk pack in Drive 0 now contains a complete copy of the contents of the OS/78 system software that resides on Device A (RL0A) of your master disk pack. If you want to make an additional backup copy of the system disk, type the following command:

```
.BO/RX
```

   then perform steps 5 through 7 again.

8. Store the master disk pack in a safe place.

2.3 DEMONSTRATION PROGRAMS

A series of programs are provided to demonstrate some of the capabilities of OS/78. These programs are run by a self-explanatory BATCH file called DEMO.BI. To initiate this demonstration, start the system and type the following command in response to the monitor's period (.):

```
.SUBMIT DEMO/H/T
```

The batch stream controlling the demonstration repeats it every three minutes until you type a CTRL/C.

Once you become familiar with the system, you may wish to remove the files comprising the demonstration to make room for other programs. To do so, use the DELETE command and type:

```
.DEL DEMO??.*
```

2.4 DEVICE AND FILE NAMES

2.4.1 Devices

The system recognizes each of its devices by one of the logical names listed in Table 2-2. You use these names in command strings to the system programs to specify the devices that you want to use. Although the system can recognize all the names, no one DECstation can have all listed physical devices. The SET HANDLER command, described in Chapter 3, allows you to specify which devices are available.

If you refer to a device that does not physically exist, the system will enter a loop condition and will not print any error message. You must restart the system by pressing the START or BOOT button to resume OS/78 operation.

Table 2-2
OS/78 Permanent Logical Device Names

Name	Description
DSK	Default output device; usually same as SYS
SYS	The diskette or disk pack inserted in Drive 0 where the monitor and system programs reside.
RL0A	Unit A of RL01 Drive 0
RL0B	Unit B of RL01 Drive 0
RL0C	Unit C of RL01 Drive 0
RL1A	Unit A of RL01 Drive 1
RL1B	Unit B of RL01 Drive 1
RL1C	Unit C of RL01 Drive 1
RXA0	The diskette inserted in Drive 0
RXA1	The diskette inserted in Drive 1
RXA2	The diskette inserted in Drive 2
RXA3	The diskette inserted in Drive 3
TTY	Keyboard/screen

(continued on next page)

Table 2-2 (Cont.)
OS/78 Permanent Logical Device Names

Name	Description
LPT	Line printer
LQP	Letter quality printer
SLU2	Serial port #2 (hard copy)
SLU3	Serial port #3 (hard copy)
VLU2	Serial port #2 (video display)
VLU3	Serial port #3 (video display)
BAT	Batch handler
VXA0	Extended-Memory Device

OS/78 recognizes each of its devices by a permanent name that is built into the system. You can temporarily assign a name of your choice, called a user-defined device, to a device name by using the ASSIGN command. For example,

```
.ASSIGN RXA1 PLACE
```

causes all future references to PLACE to address the device RXA1. More than one user-defined device name can be active at a time. The DEASSIGN command causes OS/78 to eliminate any user-defined device names. Also, all such user-assigned names are lost when you restart OS/78 with the START or BOOT button or with the BOOT command.

2.4.2 File Names and Extensions

When referring to a file-structured device, the system ordinarily expects you to specify a file name. Files are referenced symbolically by a name of up to six alphanumeric characters followed, optionally, by a period and an extension of up to two alphanumeric characters. The extension to a file name is generally used as an aid for remembering the format of a file.

In most cases, you should use the standard OS/78 file name extensions listed in Table 2-3. If an extension is not specified for an output file, some system programs append assumed (default) extensions. Where an extension for an input file is not specified, the system searches for that file name with the default extension. Failing to find such a file, the system searches for the original file without an extension. For example, if PROG were specified as an input file to PAL8, the assembler for the OS/78 Operating System, the system would first look for the file PROG.PA (since .PA is the standard extension for PAL8 input files). If PROG.PA were not found, the system would try to find the file PROG (with no extension). Some system programs do not use default extensions; refer again to Table 2-3 and to the individual system program descriptions for additional details.

Some programs also accept the characters * and ? in file names. These characters, called wildcard characters, have special meaning to the programs involved. Section 2.5.6 describes how to use them.

Since files are the basic units of the OS/78 system, an understanding of files and their structure is helpful when using OS/78. A description of files and file-structured devices is given in Section 2.8. This section can be skipped since it is not necessary for properly operating the OS/78 system, but does give an insight into what is happening in the system.

Table 2-3
OS/78 Extensions

Extension	Meaning
.BA	BASIC source file (default extension for a BASIC input file).
.BI	Batch input file (input for BATCH).
.BN	Absolute binary file (default extension for ABSLDR and BITMAP input files; also used as default extension for PAL8 binary output file).
.CM	Command file.
.DI	Directory listing file.
.FT	FORTRAN language source file (default extension for FORTRAN input files).
.HN	I/O device handler save image.
.LD	FORTRAN load module file (default assumed by run-time system, FORTRAN IV loader).
.LS	Assembly listing output file (default extension for PAL8).
.MP	File containing a loading map (used by the Linking Loader, MAP command).
.PA	PAL8 source file.
.RA	RALF assembly language file (FORTRAN IV).
.RL	Relocatable binary file (default extension for a Linking Loader input file).
.SV	Memory image file (SAVE file); default for the R, RUN, SAVE, and GET commands.
.TM	Temporary file generated by system.

OS/78 commands and the abbreviated forms for each command are summarized at the end of this chapter. Greater detail on each command is given in Chapter 3.

2.5 ENTERING MONITOR COMMANDS

OS/78 operating system commands are entered at the terminal in response to the period (.) printed by the monitor. Pressing the RETURN key initiates execution of the entered command. Some commands require a particular format to distinguish between input and output files or to name certain devices. Any errors that are made while utilizing these commands result in an error message being displayed by the monitor. Error messages are summarized in Appendix G.

2.5.1 OS/78 Command Format

An OS/78 command is usually made up of the following parts:

1. a command word or words,
2. a list of output devices and files,
3. a left-angle bracket (<),
4. a list of input devices and files,
5. a series of options, and
6. the RETURN or ESCape key.

Parts 2 through 5 above are called the command parameters because they supply information concerning the devices and files that are affected by the command.

The general format for a command may thus be expressed as follows:

```
COMMAND outspec<inspec/options (terminator)
```

where:

COMMAND	is any OS/78 command.
outspec	is the output specification. This usually consists of a device name, a colon (:), a file name, and a file name extension (Example: RXA0:TEST.PA). Output specifications are optional. If you do not specify an output device, the system assumes DSK:. If you do not specify a file name, the system assumes you are referring to any or all the files on the designated device; the exact action depending on the command.
<	is the specification separator. Specifications to the left are output specifications. Specifications to the right are input specifications.
inspec	is the input specification. This usually consists of a device name, a colon (:), a file name, and a file name extension (Example: RXA1:TEST1.PA). For each input specification that you do not explicitly name a device, the system assumes the device associated with the previous input specification. If you do not specify a device name for the first input specification, the system assumes DSK:.
/options	is one or more option specifications that modify the command; these are discussed in Section 2.5.4 and in descriptions of each of the commands in Chapter 3.
terminator	is the command line terminator. This is usually a carriage return (pressing the RETURN key). This causes the system to begin execution of the command. In some cases, you must use the ESCape key. This is explained where applicable.

The command word and parameters in each command must conform to obey the following rules:

1. The command word or words may be abbreviated as shown in Table 2-5.

2. No command may have more than three output devices or files, and these must be separated by commas. Some commands, such as COPY and DIRECT, may have no more than one output specification.

3. You can omit the left-angle bracket (<) if you do not specify any output devices or files.

4. In general, no command may have more than nine input devices or files. The DIRECT and COPY commands are limited to five.

5. Multiple input and output device or file specifications must be separated by a comma (,).

6. The file name may not contain more than six characters. The extension may not contain more than two characters.

7. In a command line, the device name is followed by a colon (:) and is always separated from any file name by a colon.

For example, the command line

```
.DIR RL0A:/E
```

when executed by pressing the RETURN key, displays the directory of RL0A on the terminal screen. The E option also causes the system to display any empty files in the order that they are located on the disk.

The command line

```
.COPY RXA0:<RL1A:ARITH.PA,MULT.PA,DIVIDE.PA
```

when executed, transfers the specified files ARITH.PA, MULT.PA and DIVIDE.PA from RL1A to RXA0.

Commands also can be simplified by the use of defaults as explained in Section 2.6.

2.5.2 Incorrect Commands

If you enter an incorrect command line and then press the RETURN key, one of the following results will occur:

1. If the command is not recognized, the system responds with a question mark followed by the monitor's period (.), and then waits for you to enter a correct command.

2. If the command is recognized, execution will begin immediately. If you see an error in the command line, press CTRL/C quickly to terminate the action and return control to the monitor. Whether this step prevents execution of the incorrect command depends on the type of command and the file(s) involved.

   You can destroy important files if you incorrectly issue any of the following commands:

   COPY SAVE SQUISH DELETE ZERO

   In fact, the error may actually destroy part of the software. Therefore, be extremely careful when using these commands. Any specific cautions that should be exercised when using these commands are fully described in Chapter 3. Always be sure to make a back-up copy of the system and all other important files that you have created (see Section 2.2).

Section 2.5.3 describes how to correct any errors that have been made in the command line prior to executing the command.

2.5.3 Correcting Errors

Until the RETURN key is pressed, the system will not process the command line. Therefore, the command line is still available for correction. Always check the command lines before pressing the RETURN key since a line with errors may cause undesirable results.

To correct typing mistakes, use the DELETE key. This key erases the last character typed. Each use of the DELETE key causes one more character to be erased. The correct character or characters can then be typed.

A command line may be deleted completely before it is entered by typing CTRL/U (produced by holding down the CTRL key and pressing the U key). This echoes as ^U and returns control to the monitor without processing the current line. The monitor's period (.) indicates that the system is ready to accept a new command. You can then retype the command line. You can use the LINE FEED key to verify the contents of a command that is being entered. The system will display the portion of the command line that you have entered.

2.5.4 Using Input/Output Options

In addition to specifying output and input files in the command line, you can specify various options to perform select certain functions. Options are numbers, alphanumeric characters, or their full-word equivalents. Numbers used as options are generally contained in the command line with the equal sign (=) or square brackets ([]) construction. The alphanumeric option characters are set off from the I/O specifications by the slash (/) character for single character options and parentheses for a string of single characters. The usage of the slash, parentheses, equal sign, and square brackets is explained below.

You will find an explanation of the single character options in the command descriptions in Chapter 3. Appendix I lists the full-word equivalents to the single character options. A group of general-purpose options known as dash options that you can use in most command lines are described below.

2.5.4.1 **Equal Sign Construction** - An equal sign (=) followed by an octal number may occur only once in a command line and must be followed by a separator character (comma or left-angle bracket), other options, or a line terminator (RETURN or ESCape). For example,

```
•DIRECT SYS:=3
```

will list the directory of the system device on the screen in three columns.

2.5.4.2 **Slash Construction** - A single alphanumeric character is preceded by a slash and can occur anywhere in the command line (even in the middle of a file name) although the usual position is at the end of the line. For example:

```
•COPY RXA1:SECOND.EX<RXA0:FIRST.EX/T
```

means the same as

```
•COPY RXA1:SECOND.EX/T<RXA0:FIRST.EX
```

The option /T instructs the monitor to assign the current date to the output file.

2.5.4.3 **Parentheses Construction** - When two or more letter options are used, you can group them together inside parentheses. This construction is valid anywhere in the command line. For example,

```
•COPY RL1B:OUTPUT.EX<SYS:INPUT.EX(QT)
```

means the same as

```
•COPY RL1B:OUTPUT.EX<SYS:INPUT.EX/Q/T
```

2.5.4.4 **Square Bracket Construction** - The square bracket construction can only occur immediately after an output file name and consists of a left square bracket ([), a decimal number between 1 and 255, and a right square bracket (]). The square bracket construction allows you to optimize file storage by specifing an upper limit on the number of blocks required by your output file. For example:

```
•PAL BINARY[19],LISTIN[200]<SOURCE
```

The output files are a file named BINARY on device DSK: having a maximum length of 19 blocks, and a file named LISTIN (only six characters are significant) on the device DSK: with a maximum length of 200 blocks. The input file is SOURCE on device DSK.

2.5.4.5 **General-Purpose Dash Options** - These options provide a shorthand method for specifying devices and files with certain commands. The form is:

```
-ex
```

where:

-ex is one of the options specified in Table 2-3.

In the following example, typing

```
.PAL TEST-T
```

will assemble TEST.PA, store the resulting binary program in TEST.BN on DSK, and display the program listing on the terminal. The -T option saves you from typing the specification:

```
,TTY:<
```

Table 2-3
General-Purpose Dash Options

Option	Meaning
-FT	Selects the FORTRAN IV Compiler if you do not use the default file extension .FT (used with the COMPILE and EXECUTE commands).
-L	Send output to LPT. For example, typing .DIR-L will list the director of the system device on the line printer (LPT).
-LS	Generate a listing file (used with the COMPILE, EXECUTE, and PAL commands). The listing file is written onto SYS: if no output device is specified and is given a .LS extension. The listing file name is the same as the file name that immediately preceded the -LS option in the command string.
-MP	Generate a memory map (used with the COMPILE, EXECUTE, and PAL commands).
-NB	Do not create a binary file (used with the COMPILE, EXECUTE, and PAL commands).
-PA	Selects the PAL8 Assembler if you do not use the default file extension .PA (used with the COMPILE and EXECUTE commands).
-T	Send output to terminal.

2.5.5 Remembering Previous Arguments

The system remembers the arguments that you use with the CREATE, LOAD, PAL, EDIT, and EXECUTE commands by storing them in a temporary file. Thereafter, when you use any of these commands without arguments, the system will use the arguments that it stored. When you again use these commands with arguments, the system replaces the currently stored arguments with the new arguments. The storage area for the EDIT command is separate from the others and its contents will not change unless you issue an EDIT command with arguments.

The following example illustrates this feature. If you entered the command:

```
.EXECUTE TEST1.PA
```

you could compile TEST.PA by issuing the command:

```
.COMPILE
```

NOTE

> The system does not remember command arguments typed if you restart the system with the START button or the BOOT command.

2.5.6 Using Wildcards

The wildcard construction is a feature that allows you write generalized file specifications when manipulating files. Wildcards allow a file name or the extension specified in a command to be replaced totally with an asterisk or partially with a question mark. This allows you to designate classes of file names or extensions. The wild characters are particularly useful when doing multiple file transfers.

You can use the wildcard construction in the file name specifications of the COPY, DELETE, DIRECT, LIST, RENAME, and TYPE commands.

The asterisk (*) is used as a wild field to designate the entire file name or extension. This is illustrated in the following examples:

TEST1.*	All files with the name TEST1 and any extension.
*.BN	All files with a BN extension and any file name.
.	All files (except when used with the DELETE command).

The question mark (?) is used as a wildcard character to designate part of the file name or extension. A question mark is used for each unspecified character that is to be matched. For example, PR?? matches all files beginning with PR that are two to four characters long. Other examples are as follows:

TEST2.B?	All files with the name TEST2 and any extension beginning with B.
TES??.PA	All files with a PA extension and any file name from three to five characters long beginning with TES.
??.??	All files with file names of two characters or less.

The asterisk and the question mark can be specified together in the same command line:

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

A specification may not contain embedded asterisks (*). For example:

```
A*B.*
```

is an illegal specification and will produce the following message:

```
ILLEGAL *
```

If you include an * or ? in a command other than COPY, DELETE, DIRECT, LIST, RENAME, or TYPE, the following message appears:

```
ILLEGAL * OR ?
```

2.5.6.1 **Wildcard Input File Specifications** - This section describes how to use wildcards when specifying input files.

For example, to display the directory entries for all files on RXA1 with file names from two to four characters long and beginning with PR, type:

```
.DIR RXA1:PR??.*
```

To display entries for all files on RL0C with extensions ending with R, type:

```
.DIR RL0C:*.?R
```

The following wildcard specifications are illegal:

```
A*.BD      The asterisk should replace the entire file name or
  or       extension.
ABC.*D
```

```
RXA?:C.BA Wildcards may not be used in device names.
```

2.5.6.2 **Wildcard Output File Specifications** - This section describes how to use wildcards when specifying output files.

You can use the asterisk in output file names, but the question mark is illegal. If no file name is specified, then *.* is assumed.

For example,

```
.DEL *.BN<*.PA
```

is legal. This command deletes any files with a .BN extension if the file and a .PA extension exist.

Another example would be:

```
.COPY RXA1:*.BK<SYS:*.PA
```

The system will copy all files with a .PA extension from the system device onto RXA1, giving them the same file name but assigning a .BK extension to each.

The question mark may not be used in output file specifications. For example:

```
.COPY RL1B:NEW.??<OLD.BA
```

is not allowed.

The wildcard construction, when used with the COPY and DELETE commands, is capable of destroying needed files if you do not use it carefully. We recommend that you follow these rules:

1. Keep a backup copy of the system and all other important files.

2. Use the /Q option when running COPY or DELETE with wildcards. This allows you to decide what to do with each file involved before it is acted upon by the system.

2.5.7 Indirect Commands (At Symbol (@) Construction)

The at symbol (@) construction allows you to include many file names and options in a single command, so that you do not have to type them each time you want to use them. You do this by creating a file with the Editor Program (EDIT command) that contains the desired file specifications and options. You then reference this file using the @ construction anywhere within the argument (file and option list) portion of a command. The form of the @ construction is:

```
@dev:file.ex
```

If dev: is omitted, DSK: is assumed. If the extension is omitted, .CM is assumed.

The system takes the text in the specified file and inserts it into the command string in place of the indirect command file reference. For example, if you create a file with the Editor called FLIST.CM that contains the string:

```
FILEB,FILEC/L,FILED
```

then the command:

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

could be replaced by the command:

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

You cannot use an indirect command file reference in arguments to the following commands:

ASSIGN	GET
BOOT	ODT
DATE	R
DEASSIGN	RUN
DUPLICATE	SAVE
FORMAT	START

Carriage returns and line feeds within the file are ignored, but nulls are not. A null signifies the end of a line. A command file may not be more than 512 characters in length.

You must separate an indirect command specification from other command specifications that follow on the same line with a non-alphanumeric character such as an apostrophe (').

2.5.8 Asterisk (*) Prompting Symbol

Under certain conditions, OS/78 may display an asterisk (*) as a prompting symbol. This display is a function of a special program called Command Decoder (described in Appendix D), and signifies a request to enter another list of files for the current system command. Except for commands where the appearance of the asterisk prompting symbol is specifically described, ignore it and return to the monitor by typing a CTRL/C.

2.6 USING DEFAULTS

You can reduce the amount of typing necessary for entering commands by taking advantage of the default ability of the system. A default condition is a command parameter that the system assumes if you do not explicitly specify one. You can use defaults to avoid typing device names repetitively.

1. For all output files and input files, the default device assumed by the monitor is DSK: and is equivalent to SYS: (that is, RXA0:, or RL0A, the diskette or disk pack in Drive 0). For example, the monitor interprets the following two file specifications identically:

   ```
   FILEA.AB,FILEB.CD<FILEC.EF

   DSK:FILEA,AB,DSK:FILEB.CD<DSK:FILEC.EF
   ```

   You can temporarily change this default with the ASSIGN command. The system will revert to the original default each time you restart the system with the START or BOOT button or with the BOOT command.

2. When a device name is explicitly given for an input file, any additional input files will default to the last explicitly specified device name. The following two file specifications are identical:

   ```
   TTY:<FILEA,FILEB,RL1B:FILEC,FILED

   TTY:<DSK:FILEA,DSK:FILEB,RL1B:FILEC,RL1B:FILED
   ```

   In this description,

   FILEA was assumed to be on DSK: because it was the first input file.

   FILEB was assumed to be on DSK: because the preceding file (FILEA) was on DSK: (same as RL0A:).

   FILEC was stated to be on RL1B:.

   FILED was assumed to be on RL1B: because the preceding file (FILEC) was on RL1B:.

3. Some of the individual commands assume special default devices. The DIRECT and TYPE commands assume the terminal as the output device. The following command line will display the directory of the diskette inserted in Drive 1 on the terminal screen.

```
.DIRECT RXA1:
```

Note that the left-angle bracket is not needed in the above command line because no output files are specified. However, commands such as CREF (cross reference program) and LIST default to the line printer. Thus, if a line printer is not available, the output of the CREF command must be sent to a file or to the terminal (using the -T Dash option). Specific cases and use of default options are discussed in Chapter 3.

2.7 WHERE TO GET MORE INFORMATION

While you are running OS/78, you may want more information about the function of certain commands, their format, and options. You can use the HELP command to obtain this information. For example:

```
.HELP COMPARE
```

will display the following information on the terminal:

```
        SRCCOM.SV

CALLING COMMANDS:
.COMPAR outdev:file.PA<indev:file1.PA,indev:file2.PA
.COMPAR outfile.PA<infile.PA,infile2.PA - - - - (Files on DSK)

SWITCHES:
/B      Compare blank lines
/C      Don't compare (slashed) comments
/S      Don't compare tabs and spaces
/T      Convert tabs to spaces on output
/X      Don't compare or print comments

ERRORS:
?0      Insufficient core
?1      Input error file 1 (or less than 2 input files)
?2      Input error file 2
?3      Output file too large
?4      Output error
?5      Can't open output file
```

If a line printer is attached to the system, you can get a hard copy printout by sending the desired HELP text to the line printer as follows:

```
.HELP LPT:<PAL,COMPARE
```

or (by using the -L dash option)

```
.HELP PAL,COMPARE-L
```

The HELP text for PAL and COMPARE will be printed on the line printer.

2.8 FILE-STRUCTURED DEVICES

A file-structured device is a random-access mass storage device that is logically divided into a number of blocks. This kind of device can read and write any desired block. A diskette and disk pack are file-structured devices, but a terminal is not.

All OS/78 file-structured devices are logically divided into 256-word blocks. Hence, blocks of 256 words are considered the standard size for OS/78. A file consists of one or more sequential (consecutively numbered) blocks. A minimum of one block per file is required, although a single file could occupy all of the blocks on a device.

By convention, OS/78 block numbers are specified in octal radix, while file lengths (number of blocks per file) are specified in decimal radix.

There are two types of file-structured devices, system devices and non-system (or files only) devices. A system device contains a copy of the OS/78 Monitor (also called the system head) on its medium and the OS/78 system programs; a non-system device does not.

The system device is also the disk drive that is accessed when you press the DECstation's START or BOOT button or use the BOOT command. If your system has an RX01 or RX02 diskette drive, the system device is Drive 0 (RXA0). If your system has RL01 cartridge disk drives only, the system device is device A of RL01 Drive 0 (RL0A).

Table 2-4 lists the OS/78 file-structured devices and their storage capacities.

Table 2-4
Storage Capacity of OS/78 File-Structured Devices

Device	Type	Size (in blocks)
RLxA	System	4018
	Non-system	4074
RLxB	Non-system	4074
RLxC	Non-system	2018
RX01	System	431
	Non-system	487
RX02	System	932
	Non-system	981
VXA0*	Non-system	128

* DECstation 88/97 systems only

2.8.1 Using RX01 And RX02 Diskettes

The RX01 and RX02 diskettes are physically identical. The RX01 is a single-density diskette and the RX02 is a double-density diskette. This means that an RX02 diskette has approximately twice the storage capacity of an RX01 diskette (see Table 2-4).

The RX01 diskette is supplied already formatted for single-density operation. You must, however, create a directory area on it with the ZERO command before copying files to it. If you are creating backup copies with the DUPLICATE command, the ZERO command is not necessary.

The RX02 diskette is a single-density RX01 diskette that you reformat for dual-density operation with the DUPLICATE command's /D option. You must, however, create a directory area on it with the ZERO command before copying files to it. If you are creating backup copies with the DUPLICATE command, the ZERO command is not necessary.

You should observe the following rules when using diskettes:

- An RX01 disk drive can use single-density diskettes only.
- An RX02 disk drive can use either single-density or double-density diskettes. The system automatically determines the density of the diskette.
- A system diskette that contains an old version of OS/78 can run on an RX01 drive only. You can use it on an RX02 drive if you want to use it for file storage only.

2.8.2 Using the RL01 Cartridge Disk Pack

The RL01 cartridge disk pack is a high-density mass storage device that has a storage capacity that is approximately ten times that of the RX02 double-density diskette (see Table 2-4).

Each disk pack consists of three logical OS/78 devices: device A, device B, and device C. Device A and B are the same size. Transfers between these devices are faster than transfers between either one of them and device C. This is because the device C is smaller than the other two and consequently occupies less of each track on the disk than the others. This increases the search time required for device C blocks.

You must format and zero each new blank disk pack with the FORMAT and ZERO commands before using it with OS/78. If you do not do this, OS/78 cannot successfully use the disk pack.

Each disk pack incorporates a bad block mapping scheme that allows OS/78 software to ignore disk blocks that become defective. Whenever a bad block error occurs, you can use the FORMAT command to make the block unavailable to OS/78. This allows you to prolong the life of your disk packs.

2.8.3 Using the VXA0 Extended-Memory Device

The VXA0 device enables you to use the extended memory above 32K provided in DECstation 88/97 systems as though it were a separate file-structured device. The VXA0 device provides high speed I/O performance similar to that of a fixed-head disk type of storage device. You refer to it in the same way as the other OS/78 devices. For example, this command

```
.COPY VXA0:SAMPLE<RXA0:SAMPLE
```

copies a program called SAMPLE into the area of memory above 32K. The command

```
.DIR VXA0:
```

produces a directory listing of the device.

2.9 FILES

A file is the fundamental storage unit of the OS/78 system. It is a uniquely named collection of data that is treated as a unit. The format of this data is unimportant; OS/78 can manipulate several standard formats, including ASCII files, binary files, and memory-image files. The important consideration is that the data forms a single and uniquely identifiable unit within the system.

2.9.1 File Directories

To maintain records of the files on a device, blocks 1 through 6 are reserved for the file directory of that device. Thus, file structured devices are also called directory devices. Entries in this directory inform the system of the name, size, anad location of each file, including all empty files and the tentative output file, if one exists. Six blocks are always allocated, though all are not necessarily active at any given time. Block zero is unused, except on the system device, in which case it contains the system bootstrap, a program that reads in the OS/78 Monitor and the system device handler (SYS).

2.9.2 File Types

Three types of files exist in the OS/78 system:

1. Empty File - An empty file is a contiguous area of unused blocks. Empty files are created when permanent files are deleted or when the ZERO command is used.

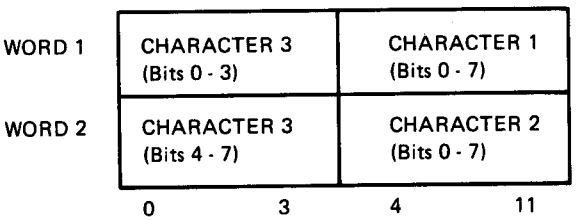
2. Tentative File - A tentative file is a file that is open to accept output and has not yet been closed. Only one tentative file can be open on any single device at one time. Tentative files never appear in directory listings.

3. Permanent File - A permanent file is a file that has been given a fixed size and is no longer expandable. A tentative file becomes permanent when it is closed.

To further understand file types, consider what occurs when a file is created. Normally, in creating a tentative file, the system first locates the largest empty file available and creates a tentative file in that space. That establishes the maximum space into which the file can expand. The user program then writes data into the tentative file. At the end of the data, the program tells the system to close the tentative file, making it a permanent file. When the file is closed, whatever space remains at the end of the file is available to contain a new file.

2.9.3 ASCII File Format

ASCII files are packed as three 8-bit characters into two words as follows:



2.10 OS/78 COMMAND SUMMARY

Table 2-5 is a summary of OS/78 commands. For each command, the table shows the command and its abbreviation printed in red ink, the format, and a brief functional description.

Table 2-5
OS/78 Command Summary

Command	Format Example	Function
ASSIGN	AS permdev user-name	Associates a new user-defined device name with the permanent logical device name.
BASIC	BAS	Requests execution of the BASIC language editor to allow for program creation and modification.
BOOT	BO/dv	Initializes the system residing on the specified system device RL or RX.
CANCEL	CA	Terminates the symbiont task (see Appendix H) that is currently running and returns OS/78 to 16K, single task operation (DECstation 78 systems only).
COMPARE	COMPA dev:output.ex< dev:file1.ex,dev:file2.ex	Compares two source files line by line and displays all their differences.
COMPILE	COM file.ex<file1.ex,.../options	Produces binary files and/or listings for specified program files. This command will chain to PAL8, BASIC, or FORTRAN IV depending on the file name extension. Under FORTRAN IV, only one input file can be specified.
COPY	COPY dev:file.ex<dev:file.ex,...	Transfers files from one OS/78 device to another. Up to five input files can be specified.
CREATE	CREA file.ex	Opens a new file for editing.
CREF	CREF file.ex	Assembles and produces a cross-reference listing from the source file. Defaults to LPT, thus the -T option must be used to output to terminal.

(continued on next page)

Table 2-5 (Cont.)
OS/78 Command Summary

Command	Format Example	Function
DATE	DA dd-mmm-yy	Enters the date into the system. If no argument is given, print current date on terminal or NONE if no date was specified.
DEASSIGN	DE	Eliminates all previous user-defined names.
DELETE	DEL dev:file.ex,.../options	Deletes one or more files. Up to five input files can be specified, separated by commas. All files must reside on the same device.
DIRECT	DIR dev:/options	Produces a listing of an OS/78 device directory.
DUPLICATE	DUP outdev:<indev:/options	Reproduces the contents of the input diskette onto the output diskette.
EDIT	ED file or ED file<file	Opens an already existing file for editing. Using the input/output construction allows a copy of the original file to be retained, while naming a new output file.
EXECUTE	EX file.ex,file.ex/options or EX file.ex<file.ex	Produces binary files and/or listings for the specified program files, loads the binary file, and executes the program. This command will chain to PAL8, BASIC or FORTRAN IV depending on the file name extension. Under FORTRAN IV, only one file can be specified.
FORMAT	FORMAT RL01	Formats RL01 disk pack for OS/78 use.

(continued on next page)

Table 2-5 (Cont.)
OS/78 Command Summary

Command	Format Example	Function
GET	GE dev:file.ex	Loads memory image files (.SV format) into memory from a device.
HELP	HE command	Displays instructional information on the specified OS/78 command.
LIST	LI dev:file.ex,...	Lists the contents of the specified files on LPT.
LOAD	LO dev:file.ex	Runs one of the OS/78 loaders for either PAL8, BASIC or FORTRAN IV depending upon the extension of the filename.
MAP	MAP dev:file.ex<dev:file.ex	Produces a memory map of the specified input file.
MEMORY	MEM n	Sets the number of memory fields available.
	MEM	Displays both the number of fields actually being used by OS/78 and the number of fields available in the hardware.
ODT	OD	Loads and starts the ODT (Octal Debugging Technique) system.
PAL	PAL dev:binfile,dev:listfile, dev:creffile<dev:infile,...	Runs the PAL8 assembler and assembles the specified source file.
QUEUE	Q filename-list/options	Line printer spooler symbiont command (see Appendix H) to queue files to be printed (DECstation 78 system only).

(continued on next page)

Table 2-5 (Cont.)
OS/78 Command Summary

Command	Format Example	Function
R	R file.ex	Loads memory image files (except OS/78 system programs) from the system device (making it equivalent to a RUN SYS file.ex command except that the system scratch area is not affected) and starts it at the starting address. If .ex is omitted, .SV is assumed.
RENAME	REN dev:new.ex<dev:old.ex/options	Changes the name of the file from the input name to the output name.
REQUEST	REQ symbiont-name	Starts symbiont task (DECstation 78 systems only. See Appendix H).
RUN	RU dev:file.ex	Loads a memory image file (except OS/78 system programs), moves its Core Control Block to the system scratch area, and starts the program at its starting address. This command is equivalent to a GET and START command, and must be used to run a program that does not reside on SYS:.
SAVE	SA dev:file.ex options	Saves an image of the program currently in memory on the device specified. If .ex is omitted, .SV is assumed.
SET		Changes device handler characteristics. ter: = TTY:, SLUx, VLUx any: = any device NO = optional modifier to inhibit effect
	SET any: [NO] READONLY	Makes device read-only device.
	SET LPT: [NO] LC	Prints lower-case characters.

(continued on next page)

Table 2-5 (Cont.)
OS/78 Command Summary

Command	Format Example	Function
SET (Cont.)	SET LPT:WIDTH n	Sets column width to n.
	SET LPT:COL n	Sets number of columns displayed by DIRECT (ter: also).
	SET ter: [NO] ARROW	Sets terminal so that CTRL characters map as ^char rather than the actual code.
	SET ter:COL n	Sets number of columns displayed by DIRECT (LPT: also).
	SET ter: [NO] ECHO	Enables character echoing on input.
	SET TTY: [NO] ESC	Sets console terminal so that ESC code maps to a $ rather than 033(8).
	SET ter:HEIGHT m	Sets screen height to m.
	SET SYS: [NO] INIT	Executes command in INIT.CM when system is bootstrapped.
	SET SYS: [NO] INIT cmd	Executes command (cmd) when system is bootstrapped.
	SET ter: [NO] PAGE	Enables CTRL/Q and CTRL/S.
	SET ter: [NO] PAUSE n	Enables pause of n seconds during output.
	SET ter: [NO] SCOPE	Deletes character from screen rather than printing a \ when DELETE key is used.
	SET ter:WIDTH n	Sets column width to n.
	SET HANDLER old<new	Replaces handler currently in system (old) with new handler (new) called: handlername.HN.

(continued on next page)

Table 2-5 (Cont.)
OS/78 Command Summary

Command	Format Example	Function
SET (Cont.)	SET HANDLER/L	Lists names of handlers currently in system.
SQUISH	SQ dev:<dev:	Eliminates all embedded empty files on the device.
START	ST ffnnnn	Starts the program currently in memory at location nnnn in field ff.
	ST	Starts the program currently in memory at the starting address specified in the Core Control Block.
SUBMIT	SU dev:file.ex<dev:file.ex	Runs the BATCH program where the output is the optional spooling file and the input is the BATCH input file.
TERMINATE	TER	Causes OS/78 to terminate its operation and to run a user-created stand-alone program called TERMIN.SV.
TYPE	TY dev:file.ex,...	Displays the specified files on the terminal.
UA	UA (argument)	Remembers the argument where the argument must be a legal OS/78 command. UA with no argument executes the last remembered argument.
UB	UB (argument)	Similar to UA.
UC	UC (argument)	Similar to UA.
ZERO	ZERO dev:/options	Zeros the specified device directory, deleting any files that may exist or the device.

CHAPTER 3

OS/78 COMMANDS

OS/78 system software supports a set of commands that allow you to direct computer operations. Each OS/78 command has a standard format that you should be familiar with before using the command.

The format for each command is shown, including one or more examples. Each command description is followed by any options that you can use and the error messages that may result when using the command.

You must begin each command on a new line. Enter the command and its arguments immediately to the right of the period (.), the monitor's prompt symbol. Press the RETURN key to enter the command into the system for execution. Characters that are system responses are shown in black; characters that you type are shown in red.

Commands may or may not require arguments. Many of the commands have default options that are automatically inserted in the command line by the system if an argument is not explicitly specified. Default options may also vary with each command. The use of defaults is described in Section 2.6. The full-word equivalents for each option are listed in Appendix I.

ASSIGN

3.1 ASSIGN COMMAND

The ASSIGN command associates a user-defined name with one of the permanent logical device names listed in Table 2-1.

Format:

```
ASSIGN permdev user-name
```

Example:

```
.AS RXA1 DEV2
```

In this example, device RXA1 is assigned the symbolic name DEV2 and can now be addressed by both RXA1: and DEV2:. Note that the colon is not used with both the permanent device name and the user-defined name in the ASSIGN command. However, when referenced in most other commands the colons must be used with the user-defined name.

If a user-defined name is not specified in the command line, any one that may have existed is removed and only the permanent device name is valid.

Example:

```
.AS RXA1 DEV2     -Previous ASSIGN command
.AS RXA1          -User-defined name becomes invalid
```

The user-defined name that is assigned to a permanent device can be from one to four alphanumeric characters long.

NOTE

A system error may occur if by coincidence the internal representation of a user-defined name matches some other internal code used by the system.

The ASSIGN command is particularly useful if an LQP line printer is attached to the system instead of an LA78 line printer. Since several of the OS/78 commands automatically send output to the LA78 line printer (LPT), assigning LPT to LQP allows output to be sent to the LQP78 line printer.

Example:

```
.AS LQP LPT
```

ASSIGN processing is done by the monitor.

BASIC

3.2 BASIC COMMAND

The BASIC command requests execution of the BASIC editor.

Format:

```
BASIC
```

Example:

```
.BASIC
NEW OR OLD--
```

For additional information on BASIC, refer to Chapter 6.

This command runs CCL.SV and BASIC.SV programs.

BOOT

3.3 BOOT COMMAND

The BOOT command allows DECstation 88/80 users to start (bootstrap) the OS/78 system that resides on another disk device.

Format:

BOOT/dv

where:

dv — is the first two characters of the name of the system device (RX for RX02 and RL for RL01) where you want to begin OS/78 operation.

Example:

.BOOT/RX

In this example your current system device is the RL01 disk (RL0A). Typing the above command causes the system to read in the monitor from the system diskette mounted in the RX02 disk drive and use it as the system device.

This command causes execution of the CCL.SV and the BOOT.SV programs.

CANCEL

3.4 CANCEL COMMAND

The CANCEL command terminates symbiont task operation and returns OS/78 to stand-alone operation with 16K of memory.

Format:

```
CANCEL
```

This command can be used on DECstation 78 series systems only. Refer to Appendix H for more information.

COMPARE

3.5 COMPARE COMMAND

The COMPARE command compares one source input file to another source input file to see if they are identical or different. Differences are sent to the output file.

Format:

```
COMPARE outdev:file.ex<indev:filel.ex,indev:file2.ex/options
```

where:

outdev:file.ex	is the specification of the file containing the results of the comparison and the device (dev:) where you want reside.
indev:filel.ex	is the first input source file for comparison
indev:file2.ex	is the second input source file for comparison
/options	is one or more of the options listed in Table 3-1

Table 3-1
COMPARE Options

Option	Meaning
/B	Consider a blank line as a valid input line containing blanks instead of a carriage return.
/C	Ignore all comment fields during the comparison (assumes assembly language source files are being compared).
/S	Ignore all tabs and spaces during the comparison.
/T	Convert all tabs from the input file to spaces on the output device.
/X	Ignore all comment fields during the comparison and do not send any comments to the output device.

NOTE

When you use the /S and /B options together, blank lines are ignored.

Along with the necessary arguments for the command line are COMPARE options that can restrict or enhance the result of the COMPARE operation. The COMPARE program is commonly used to print all the editing changes between two different versions of a program. This makes it a useful documentation and debugging tool.

COMPARE makes the following assumptions:

- If you omit an input device, COMPARE assumes DSK
- If you omit the output specification altogether, COMPARE assumes TTY. (In most cases you will want to see the results on the terminal.)

Example:

```
.COMPA  RL0B:DIFVER<RL0C:VERS1.FT,VERS2.FT
```

In this example, every source line in files VERS1.FT and VERS2.FT on RL0C is compared and the results of the comparison are sent to the file named DIFVER on device RL0B. The file extension .LS is automatically added to DIFVER.

Before the first two lines of input are compared, the current version number of the COMPARE command is sent to the output device followed by a header for each input file. The header, the first line in the file, is usually the title, and date, file name, or a description of that file.

Example:

```
SRCCOM V5A

1)  /THIS IS THE FIRST SOURCE COMPARE FILE

2)  /THIS THE SECOND SOURCE COMPARE FILE
```

The system performs the compare in two passes: the first compares file 1 to file 2 and the second compares file 2 to file 1.

When the COMPARE command is executed, lines from the first file that are either identical to or different from lines in the second file are sent to the output device until there are three consecutive lines that agree. The first line sent to the output device is preceded by the numeral one, a closing parenthesis, and a 3-digit decimal number. The numeral one represents the first file specified in the command line and the 3-digit decimal number is the page that the difference group is located on. Each succeeding line is also preceded by the numeral one and closing parenthesis. When three lines agree, only the first line is sent to the output device. The next line consists of a group of asterisks (*). This ends the first difference group for file 1.

Example:

```
1)002   C
1)       D
1)       E
1)       F
1)       G
1)       H
1)       I
1)       J
1)       K
1)       L
1)       M
1)       N
****
```

Now the command proceeds to find the first difference group for file 2. It sends all lines from the second file that both agree and disagree with lines from the first file to the output device until there are three consecutive lines that agree.

NOTE

To change the number of consecutive lines that interrupt processing, use the =k option. By specifying the =k option, you can use any octal number to set the number of consecutive lines that separate one difference group from another.

The first line sent to the output device is preceded by the numeral two, a closing parenthesis, and a 3-digit decimal number. The numeral two represents the second file specified in the command line and the 3-digit decimal number is the page the difference group is located on. Each succeeding line is also preceded by the numeral two and closing parenthesis. When three lines agree, only the first line is sent to the output device. The next line consists of a group of asterisks (*). This ends the first difference group for file 2.

Example:

```
2)002    4
2)       6
2)       2
2)       X
2)       B
2)       Y
2)       S
2)       F
2)       M
2)       A
2)       R
2)       N
****
```

If all the lines on both files agree, a message "NO DIFFERENCES" is sent to the output device. (Header lines are disregarded.) (See Table 3-2 for error messages.)

3.5.1 Using COMPARE's Options

The /C option compares all source lines in one file with source lines in another file but disregards all assembly comment lines. In addition to disregarding comment lines in a comparison, you can restrict the output of the comment lines by using the /X option. When you specify the /X option, all comments are ignored while the comparison takes place and comment lines are not sent to the output device.

If the format of one file is different from that of another because of tabs and spaces, use the /S option. When you specify the /S option, the COMPARE command compares all source lines but disregards all tabs and differing spaces.

Blank lines are normally disregarded when comparing two files. If, in a comparison, a blank line is to be considered as a valid input line instead of a carriage return, use the /B option. The /B option causes a blank line to be considered a valid line containing all blanks.

This command runs the CCL.SV and SRCCOM.SV programs.

Table 3-2
COMPARE Error Messages

Message	Meaning
?0	The differences between the two files are too numerous for an effective compare. If the large number of differences is caused by comment lines, use the /X option to eliminate the comment lines.
?1	File 1 has an input error, or two input files are not specified in the command line.
?2	File 2 has an input error.
?3	The output device ran out of room.
?4	The output has an error.
?5	An output file cannot be created.

COMPILE

3.6 COMPILE COMMAND

The COMPILE command assembles PAL source input files to produce absolute binary output files, compiles FORTRAN source input files to produce relocatable binary output files and compiles, loads, and executes BASIC source input files.

Format:

```
COMPILE outdev:file.ex<indev:file.ex/options
```

Along with the necessary arguments for the command line are several PAL, BASIC, and FORTRAN options that can modify the result of the COMPILE operation. For detailed information on PAL, BASIC, and FORTRAN command formats and options, see the appropriate language chapter.

When the COMPILE command assembles a PAL input file, the absolute binary output file created usually has a .BN extension (the default for PAL8). When the COMPILE command compiles a FORTRAN input file, the relocatable binary output file created usually has an .RL extension (the default for FORTRAN IV). Both absolute and relocatable binary output files may be used as input files in the LOAD command.

An absolute binary file is an independent stand-alone program; a relocatable binary file is a dependent program that needs other system programs to aid it in its execution.

The extension of the first input file determines the compiler to be used. The .PA uses PAL8, .FT uses FORTRAN, .BA uses BASIC. If the extension is not specified as part of the file name in the command line, all assembler and compiler extensions are compared with the unspecified extension until a match is made. (The search order is PAL8 (.PA), FORTRAN (.FT), then BASIC (.BA).) The appropriate assembler or compiler is called and executed. If you use an extension other than the ones listed in Appendix E, specify a general-purpose dash option (see Section 2.5.4.5) to specify the appropriate compiler or assembler.

Example:

```
.COM RXA1:COMP<ACCTS.03-FT
```

In this example, the file TEST.FT resides on DSK (default device). Since no extension was given in the command, the system searches for any file named TEST.* (wildcard extension). When it finds TEST.FT, it runs the FORTRAN compiler, which produces the object file TEST.RL (default name and extension).

If no output file is specified, the first input file name specified in the command line is used as the default name of the output file.

Example:

```
.COMPILE TEST
```

The output file produced is named TEST.RL.

The COMPILE command can recall arguments from a previous COMPILE, LOAD, EXECUTE, or PAL command. Consequently the COMPILE command can be typed without any arguments until you restart the system with the START or BOOT button or with the BOOT command.

NOTE

Refer to the appropriate language chapter for any error messages received as a result of the COMPILE command.

This command causes execution of the CCL.SV program and the PAL8.SV, F4.SV, or BCOMP.SV programs and optionally the RALF.SV, LOAD.SV, and ABSLDR.SV programs.

COPY

3.7 COPY COMMAND

The COPY command transfers files from one device to another and allows you to change a file's name. (COPY command options are in Table 3-3; error messages are in Table 3-4.)

Format:

```
COPY outdev:file.ex<indev:file1.ex...file5.ex/options
```

When the COPY command is executed, all specified input files are copied onto the output device. A message listing the name of each file copied is displayed until copying is done.

Example:

```
.COPY RL1A:<PINE.SS,TEAK.,ROSE.PA

Files Copied:
TEAK.SS
PINE.
ROSE.PA
```

The three files are copied from the system device (SYS) to RL1A.

3.7.1 File Transfer

When the COPY command is executed, each input file is copied in its exact format to the output device in the same order they are found on the input device. No changes are made during file transfer and as a result, any file, whether memory image, binary or ASCII, can be transferred. During the transfer, the original creation date is transferred and included in the directory. Along with the necessary arguments for the command line, COPY options can be specified that can change and enhance the result of a file transfer.

If input files are to be transferred to the output device in the exact order specified in the command line, use the /U option.

If the majority of the input files on the input device are to be transferred to the output device, use the /V option and specify the files that are not to be transferred. The COPY command then transfers all the files that are not specified in the command line. Wildcards (described in Section 2.5.6) can be used with the COPY command.

3.7.2 Examples of COPY Commands

The following are legal COPY command strings. When the operation is completed, control returns to the monitor. Some of the examples show the use of wildcards.

Example 1:

```
.COPY RXA1:<A,B,C,D,E
```

This command string transfers the files A, B, C, D, and E from the system device (SYS) to RXA1.

Example 2:

```
.COPY LPT:<*.FT,*.BA/U
```

This command string lists all FORTRAN files, then all BASIC files on the line printer (same as LIST).

Example 3:

```
.COPY LPT:<RXA0:*.SV,*.BN/V
```

This command string copies from RXA0 to the line printer all files other than memory image (.SV) and binary (.BN) files.

3.7.3 Predeletion

Before the COPY command copies the designated input files to the output device, the output device is checked for files having the same name as the input files. If an identical file name is found, that file is deleted before file transfer is performed. This process is known as predeletion and, by default, is automatically done each time the COPY command is executed.

If there are no files with the same name on the output device, each file is tranferred to the smallest available empty space that it can fit in.

Predeletion is advantageous because it creates space for a new file on the output device by deleting the old file before file transfer. In many cases, the output device has received as many files as it can hold and therefore may not have enough room for a new tentative file. By first deleting the old file, a sufficient amount of space could be created for the new file. Predeletion normally places the new file in the space occupied by the file being replaced. Therefore, if the length of the input file is the same or shorter than the length of the deleted file, that empty space is filled. Otherwise, the new tentative file is placed somewhere else on the device, leaving a gap where the old file was deleted. This alters the order of the files on the output device after transfer.

If predeletion is not desired, use the /N option in the command line. When the /N option is specified, postdeletion takes place.

3.7.4 Postdeletion

If the /N option is used with the COPY command, the designated input files are copied to a tentative file on the output device without checking for files having the same name. When the transfer operation is completed, the tentative file is closed. Closing a tentative file makes it a permanent file and, at the same time, deletes any old files having the same name. This process, known as postdeletion, takes place only when the /N option is specified.

Example:

```
.COPY RL0B:<RL0C:ACCT06.AC/N
```

A file named ACCT06.AC on device RL0C is copied to a tentative file on device RL0B. After file transfer is completed, the file is closed and any other file on device RL0B with a filename of ACCT06.AC is deleted.

The primary advantage of postdeletion is that if an input/output error (due, for example, to a faulty disk media) occurs on reading an input file, the corresponding output file will be preserved instead of deleted. Always use the /N option when copying important files to backup diskettes to avoid losing both file and backup file in the event of an input/output error.

3.7.5 Considering the Date on a Transfer

When files are transferred from one device to another, their creation dates are also transferred. These dates, along with their new or original file names, are included in the output directory. If it is desired to assign the current date to an output file and disregard the creation date, use the /T option. This option allows the COPY command to transfer files from one device to another and append the file names and the current date to the output directory.

To transfer all input files that have the current date, use the /C option. When the /C option is used, the COPY command copies only those files that were created or changed on the current date and appends the filenames and their creation dates to the output directory.

However, the transfer of all input files that have a creation date other than the current date is done by using the /O option. By specifying the /O option in the command line, the COPY command copies only those files with a date other than the current date from one device to another, and adds the file names and their creation dates to the output directory.

3.7.6 File Protection During a Transfer Operation

When transferring a large number of files, use the /F (failsafe) option. When the /F option is specified and an input file is encountered which will not fit on the output device, a message is printed on the terminal informing you to dismount the current disk pack or diskette and mount a new one on the same drive. To continue the COPY operation, type any character. If the /F option is not specified, then files which do not fit on the output device are not copied. Instead, a warning message is printed:

```
NO ROOM, SKIPPING filename
```

Example:

```
.COPY RXA1:<RXA0:*.FT/F


Files Copied:
VERS09.FT
VERS12.FT
VERS10.FT
VERS11.FT
VERS7.FT

Mount Next Output Volume:

VERS5.FT
VERS6.FT
VERS8.FT
```

Before the /F option is used, all output directories on new devices that might be used for continued COPY operations should be emptied (zeroed). Empty devices by using the ZERO command before the COPY operation begins. You can continue COPY operations to a device with already existing files. However, if the length of the next file to be transferred exceeds the lengths of the empty spaces on the output device, another message for a new device to be mounted is printed on the terminal.

If many files are being transferred at one time, especially ones with similar names, each affected file specified in the command line can be verified as the correct file for the operation by using the /Q (query) option. When the /Q option is specified, the first input file name followed by a question mark (?) is printed on the terminal. At this time, decide whether or not the file is the correct one. If yes, type a Y, and that file will be transferred to the output device. If no, type any other character and that file will be ignored. After your file is either transferred or ignored, the next input file name followed by a question mark is printed on the terminal. This questioning continues until all affected files in the command line have been processed.

Example:

```
.COPY SYS:<RLOB:*.PA/Q

Files Copied:
MAIN.PA?/Y
SUB1.PA?/N
SUB2.PA?/Y
LIB1.PA?/N
LIB2.PA?/Y
```

In this example, the files named SUB1.PA and LIB1.PA are not transferred.

3.7.7 Terminating COPY Operations

The only control character used to abort COPY operations is CTRL/C.

When you type CTRL/C during a COPY operation, file transfer terminates immediately and control returns to the monitor. Because file transfer is terminated before a successful completion, some requested files may not have been transferred.

Example:

```
.COPY RXA1:<RXA0:TABLE.FT
^C
FILES COPIED:

.
```

This command executes the CCL.SV and FOTP.SV programs.

Table 3-3
COPY Options

Option	Meaning
/C	Transfers all input files that have the current date to the output device.
/F	Displays the message: MOUNT NEW DEVICE: on the terminal when a file will not fit on the output device. Dismount current device and mount new device on same unit. Continue the COPY operation by typing any character. The new device must have a valid OS/78 directory on it.
/N	Transfers each file as a tentative file, then deletes any existing file having the same name and makes the tentative file permanent. This is postdeletion.
/O	Transfers only those input files with dates other than the current date.
/Q	Displays the name of each file and pauses to allow you to select (type Y) or reject (type any other character) the transfer operation for this file.
/T	Transfers all specified input files to the output device and at the same time, change the creation date on the output files to the today's date.
/U	Finds and transfers the designated input files in the exact order they are specified in the command line, not the order they are found on the input device.
/V	Transfers all files on the input device that are not specified in the command line to the output device.

NOTE

All output directories on devices that could be used for continued COPY operations (/F) must have valid OS/78 directories on them. Use the ZERO command to create as many blank disks or diskettes as you require for the operation.

Table 3-4
COPY Error Messages

Message	Meaning
BAD INPUT DIRECTORY	The directory on the specified input device does not exist or is not valid.
BAD OUTPUT DEVICE	A non-file structured device is specified as the output device.
BAD OUTPUT DIRECTORY	The directory on the specified output device does not exist or is not valid.
ERROR ON INPUT DEVICE SKIPPING (file name)	The file specified is not transferred, due to a disk-reading error, but any previous or subsequent files are transferred and indicated in the new directory.
ERROR ON OUTPUT DEVICE SKIPPING (file name)	The file specified is not transferred, but any previous or subsequent files are transferred and indicated in the new directory.
ERROR READING INPUT DIRECTORY	A disk-reading error occurred on the input device. No files are transferred.
ERROR READING OUTPUT DIRECTORY	A disk-reading error occurred on the output device. No files are transferred.
ERROR WRITING OUTPUT DIRECTORY	A disk-writing error occurred on the output device. The device's directory is likely to be unusable.
ILLEGAL*	An * was entered as an embedded character in a file name, e.g., TMP*,BN (see Section 2.5.6 on wildcards).
ILLEGAL?	A ? was entered in an output specification (see Section 2.5.6 on wildcards).
NO FILES OF THE FORM xxxx	No files of the form (xxxx) specified were found on the current input device group.
NO ROOM, SKIPPING (file name)	No space is available on the output device to perform the transfer. Predeletion may already have occurred.
SYSTEM ERROR-CLOSING FILE	Bad directory format or system head has become corrupted.

CREATE

3.8 CREATE COMMAND

The CREATE command calls and runs the OS/78 Editor to allow a new file to be opened. Once opened, the new file is ready to receive any text typed in. Only one output file specification is allowed.

Format:

```
CREATE outdev:file.ex/options
```

After you press the RETURN key, the Editor displays the number sign (#), its prompt symbol, on the terminal to indicate that it is ready to receive commands. (See Chapter 4 for a detailed explanation of the OS/78 Editor and its commands and options.)

Example:

```
.CREATE RL0B:RUN1.PA
#
```

In this example, the CREATE command opens a file named RUN1.PA on output device RL0B.

Each time a CREATE command is executed, the accompanying arguments (device and file name) are stored in a temporary area for subsequent use by the EDIT command. This allows you to type the EDIT command without any arguments to continue editing a file that you have previously created. These arguments are not available after you restart the system with the START or BOOT button or with the BOOT command. The EDIT command cannot recall file names that have no date assigned. Therefore, use the DATE command prior to the CREATE command.

This command causes the execution of both the CCL.SV and EDIT.SV programs.

CREF

3.9 CREF COMMAND

The CREF command assembles a PAL8 program and produces a cross-reference listing. (CREF commands are in Table 3-5; error messages are in Table 3-6.)

Format:

```
CREF outdev:file.ex<indev:file.ex/options
```

Example:

```
.CREF PROGA
```

The cross-reference listing contains the source program with the assembled instructions followed by the cross-reference table, which contains every user-defined symbol and literal. These listings aid in writing, debugging and maintaining assembly language programs. Using CREF is the same as using the PAL command with the /C option specified. If no output device is specified, LPT is the default output device.

This command executes the PAL8.SV, CCL.SV, and CREF.SV programs.

Table 3-5
CREF Options

Option	Meaning
/E	Do not eliminate the intermediate file CREFLS.TM that is output from the assembly and used as input to produce the CREF listing.
/M	Cross-reference mammoth files in two passes. Pass 1 processes the symbols from A through LGnnnn; pass 2 processes the symbols from LHnnnn through Z and literals. This permits very large files to be cross-referenced.
/P	Disable pass 1 listing output. The output is reenabled when $ is encountered. Thus the $ and symbol table are printed if the /P option is used.
/U	Disable pass 1 listing output and the symbol table.
/X	Do not process literals. For programs with too many symbols and literals for CREF, this option may create enough space for CREF to operate.

Table 3-6
CREF Error Messages

Message	Meaning
2045 REFS	More than 2044 (decimal) references to one symbol were made.
CLOSE FAILED	CLOSE on output file failed.
ENTER FAILED	Entering an output file was unsuccessful - possibly output was specified to a read only device.
HANDLER FAIL	This is a fatal error on output.
INPUT ERROR	A read from the input device failed.
OUT DEV FULL	The output device is full (directory devices only).
SYM OVERFLOW	More than 896 (decimal) symbols and literals were encountered during a major pass. Try again using the /M option.

DATE

3.10 DATE COMMAND

The DATE command lets you set and inspect the current system date. You should always set the date wherever you start the system with the START or BOOT button or with the BOOT command. The system appends the current date to all files that you create, rename, or copy and prints it in directory listings and program output listings.

Format:

```
DATE dd-mmm-yy
```

where:

dd is a 2-digit decimal number representing the day,

mmm is a 3-character alphabetic string representing the month which must be the first three letters of the month, and

yy is a 2-digit decimal number representing the last two digits of the year.

NOTE

The only directory entry dates that the system can read are those for the current year and seven years preceding the current year. Any earlier dates will be printed incorrectly. If you want to examine directories with earlier dates, do not set the date when you start the system.

Example:

```
.DATE 1-MAR-79
```

If the DATE command is typed without any arguments after the date has already been entered by a previous DATE command, the current day of the week and date are displayed.

Example:

```
.DA 1-JUN-77
.DA
THURSDAY MARCH 1, 1979
```

If the date is specified incorrectly, an error message is displayed.

Example:

```
.DA 23-05-79
BAD DATE
```

This command executes the CCL.SV program and is processed by the monitor.

DEASSIGN

3.11 DEASSIGN COMMAND

The DEASSIGN command removes any user-defined names that are currently assigned to the permanent devices.

Format:

```
DEASSIGN
```

Example:

```
.AS SYS DEV1
.DE
```

In this example, the system device can no longer be referenced as DEV1. Only SYS, DSK, RXA0, or RL0s are valid.

The monitor performs the DEASSIGN function.

DELETE

3.12 DELETE COMMAND

The DELETE command deletes all specified files. (DELETE command options are in Table 3-8; error messages are in Table 3-9.)

Format:

```
DELETE indev:file1.ex...file5.ex/options
```

In response to the monitor's prompt (.), type the names of the files that you want to delete, then press the RETURN key. This will delete the files from the specified device. Wildcards are permitted. Only one device may be specified.

Example:

```
.DEL RLOB:REPORT.FT,SECUR.FT,ASSIST

Files Deleted:
SECUR.FT
REPORT.FT
ASSIST
```

A message followed by the name of each file deleted is displayed on the terminal.

CAUTION

If you do not specify any filenames, *.* (meaning all files) is the default input specification.

This command deletes all copies of your system FORTRAN files that are on RXA1.

Example:

```
.DEL RXA1:*.*<SYS:*.FT
```

An advanced format that can be used with the DELETE command is:

```
DELETE outdev:file.ex<indev:file 1,.../options
```

If an output specification is given, then this command works differently. Instead of deleting the input files specified, the command lists those file names that match the given file specification, and that are on the input device. This list is then transformed to an output list by applying the wildcards specified in the output specification. The resulting list is then used to delete files from the output device specified.

This command executes the CCL.SV and FOTP.SV programs.

Table 3-7
DELETE Options

Option	Meaning
/C	Delete only those specified files having the current date.
/N	Displays the names of all files on the output device that match the specification, but does not delete them.
/O	Delete only these specified files with dates other than the current date.
/Q	Display each file name on the terminal; type a Y (for yes) to delete the file or any other character (for no) if the file is not to be deleted.
/V	Delete all files that are not specified in the command line.

Table 3-8
DELETE Error Messages

Message	Meaning
BAD OUTPUT DEVICE	This message usually appears when a non-file structured device is specified as the output device.
BAD OUTPUT DIRECTORY	The directory on the specified output device is not a valid OS/78 device directory.
DELETES PERFORMED ONLY ON INPUT DEVICE GROUP 1 CAN'T HANDLE MULTIPLE DEVICE DELETES	More than one input device was specified with the /D option when no output specification (device or file name) was included.
ERROR READING OUTPUT DIRECTORY	Self-explanatory.
ERROR WRITING OUTPUT DIRECTORY	Self-explanatory.
ILLEGAL*	An * was entered as an embedded character in a file name, for example, TMP*.BN.
ILLEGAL?	A ? was entered in an output specification.
NO FILES OF THE FORM xxxx	No files of the form (xxxx) specified were found on the current input device group.

DIRECT

3.13 DIRECT COMMAND

The DIRECT command sends the directory listing for file-structured devices to an output device. The terminal is the default output device. (DIRECT command options are in Table 3-9; error messages are in Table 3-10.)

Format:

```
DIRECT outdev:listfile<indev:filetype,.../options
```

Example:

```
.DIRECT RXA1:
```

Along with the necessary arguments for the command line are the DIRECT options (shown in Table 3-9) that control the DIRECT operation.

Table 3-9
DIRECT Options

Option	Meaning
/B	Include the starting block numbers (octal) for each file in the directory listings.
/C	List only files with the current date, that is, the date entered with the most recent DATE command.
/E	Include empty file spaces in the directory listing.
/F	List only the file names and omit the lengths and dates.
/M	List only the empty files from the directory.
=n	List the directory in the number of columns specified by n. The n is a number ranging from 1-7.
/O	List only those file names from the directory with dates other than the current date.
/R	List the file name specified in the command line and the names of any files stored on disk following that specified file name.
/U	List each input specified as a separate directory listing with the file lengths and dates in the exact order specified in the command line.
/V	List all the files on the input device that are not specified in the command line.

The standard directory listing has the following format:

filename.ex	nnn	dd-mmm-yy
(filename with extension)	(number of blocks used in decimal)	(day, month, year)

NOTE

If the date is not entered into the system via the DATE command, directory listings will not contain file creation dates.

If no output device is specified, the output defaults to the terminal. However, if you want the directory printed on LPT, use the -L general-purpose dash option in the command line.

Example:

```
.DIR RLOC:-L
```

is equivalent to

```
.DIR LPT:<RLOC:
```

Wildcards are permitted in the input specification but are not allowed in the output specification.

Example:

```
.DIR SYS:*.FT
```

As a result, only those files with FT extensions in the directory are displayed on the terminal.

Example:

```
.DIRECT RXA1:WN????.*
```

will display all the files from the directory on RXA1 that have a file name two to six characters long beginning with WN, and having any extension.

Example:

```
.DIR RXA1:DIRECT.DI<SYS:
```

saves the directory listing of the system device and stores it in a file named DIRECT.DI on device RXA1.

3.13.1 Directory Listing Selected by Date

Use the /C option to display the names of the files that have the current date on the terminal. To display only those file names from the directory with a date other than the current date, use the /O option in the command line.

3.13.2 Directory Format Selection

To display just the file names from the directory, use the /F option in the command line. The /F option displays only the file names and omits all block lengths and dates.

The /E option displays file names with their extensions, lengths, and dates, and empty files with their lengths in the exact order that reside on the device.

When the /M option is specified, each empty file and its file length are displayed. Do not mix the /M option with the other options.

The =n option displays the list of file names in the number of columns specified by n. The value of n must be in the range 1-7 inclusive. The values 1 through 7 can be substituted for n. For use on the terminal, values 1 through 3 are suggested. Values more than 3 are normally used for the line printer. Two columns is the default value. You can change this value with the SET command.

To display a specific file name and all file names from that point on, use the /R option in the command line. This option should not be mixed with other options.

If there are a large number of file names from the directory to be displayed, use the /V option. When you specify the /V option, all file names that are not specified by the command line are displayed.

The starting block number (octal) of each file in the directory is displayed by using the /B option. The starting block numbers are in the form of nnnn and displayed following the file name and extension.

This command causes the execution of both the CCL.SV and DIRECT.SV programs.

Table 3-10
DIRECT Error Messages

Message	Meaning
BAD INPUT DIRECTORY	This message occurs when the directory is non-existent or not in legal format.
DEVICE DOES NOT HAVE A DIRECTORY	This input device is a non-directory device, for example, TTY or LPT.
EQUALS OPTION BAD	The =n option is not in the range 1-7.
ERROR CLOSING FILE	System error.
ERROR READING INPUT DIRECTORY	A disk-read error occurred while trying to read the directory.
ERROR WRITING FILE	A disk-write error occurred while trying to write the output file.
ILLEGAL*	An asterisk (*) was included in the output file specification or an illegal * was included in the input file name.
ILLEGAL?	A question mark (?) was included in the output file specification.
NO ROOM FOR OUTPUT FILE	The output device does not have sufficient space for the directory listing file to be written.

DUPLICATE

3.14 DUPLICATE COMMAND

The DUPLICATE command copies the entire contents of one diskette to another diskette. If your system has an RX02 diskette drive, the command allows you to format diskettes for single or double-density operation. (DUPLICATE command options are in Table 3-11; error messages are in Table 3-12.)

NOTE

You cannot use this command to duplicate RL01 disk packs. Use the procedures described in Section 2.2.

Format:

```
DUPLICATE outdev:<indev:/options
```

where:

outdev: is the output device (RXA0 thru RXA3).

indev: is the input device (RXA0 thru RXA3).

/options is one or more nonconflicting options listed in Table 3-11.

If, in a transfer operation, the density of the output diskette is different from that of the input diskette, DUPLICATE automatically reformats the output diskette to the same density as the input diskette before proceeding. DUPLICATE can use up to 16K of memory for copy operations. To obtain the maximum copying speed, use the MEMORY command to provide the necessary memory space.

Example:

```
.DUPLICATE RXA1:<RXA0:
```

The contents of input device RXA0 is copied onto output device RXA1.

NOTE

If your system has both RX01 and RX02 drives, you can duplicate between them but only with single-density diskettes.

3.14.1 Changing Devices Before and After Executing the DUPLICATE Command

Since the monitor resides on the system device, the system device must remain on line when interacting with the monitor and any OS/78 system programs.

To duplicate the contents of a diskette containing only files (one that does not contain a system head), and your system has only two diskette drives, use the /P option. This causes the system to pause before and after the selected operation. A ready message followed by a question mark is displayed. This pause provides time to remove the system diskette and mount a new diskette for the actual duplication. To start the DUPLICATE operation, type a Y after the question mark and press the RETURN key. After the DUPLICATE operation is completed, a message is displayed asking if the diskette containing the monitor is remounted. The second pause provides time to remove the new copy and remount the system diskette so control can return to the monitor. After remounting the system diskette, type a Y after the question mark and press the RETURN key. Control returns to the monitor.

Example:

```
.DU RXA0:<RXA1:/P
READY?Y
IS MONITOR REMOUNTED?Y
.
```

3.14.2 Performing a Read Check

To check the integrity of a diskette, use the /R option and specify only the input device. When you specify the /R option, every block of the specified device is read and checked for bad sectors. If bad sectors exist, messages indicating the device, track number and sector number are displayed. If there are no bad sectors, control returns to the monitor.

Example:

```
.DU <RXA1:/R
INPUT DEV READ ERROR TRACK:nn SEC:nn
```

NOTE

Bad sectors are usually a result of medium wear or mistreatment. Move your files onto a new diskette rather than using one with bad sectors.

3.14.3 Transfer Without Checking for Identical Contents

To transfer the contents of one device to another without performing any checks, use the /N option. When you specify the /N option, the contents of the input device is transferred or copied to the output device. If /N is not specified, each block on the output device is compared to that of the input device to assure accuracy.

3.14.4 Check for Identical Contents Without Transferring

To check whether the contents of diskettes are identical without transferring, use the /M option. The /M option reads and checks the contents of both devices for identical contents. If they do match, control returns to the monitor. However, if they differ in any way, a message, containing the device name, the track number, and the sector number of the sectors or blocks that do not match is diplayed.

Example:

```
.DU RXA1:<RXA0:/M
COMPARE ERROR TRACK nn SECTOR nn
```

3.14.5 Formatting Diskettes (RX02 Drives Only)

The /S and /D options format diskettes for single-density or double-density use. To format a diskette, enter its name by itself in the command line, followed by the option. (RXCOPY considers a device name entered by itself to be an output device.)

For example, the following command sequence reformats the diskette in drive 1 from single density to double density.

```
.DU RXA1:/D
```

To change it back to single-density, type

```
.DU RXA1/S
```

This command causes the execution of both the CCL.SV and the RXCOPY.SV programs.

Table 3-11
DUPLICATE Options

Option	Meaning
/D	Format the output diskette for double-density use (RX02 drives only).
/M	Check both devices for identical contents and list the areas that do not match but do not perform a transfer unless otherwise specified.
/N	Copy the contents of one device to another but do not check them for identical contents unless otherwise specified.
/P	Pause and wait for user response before and after data transfers to/from diskette. To continue, type a Y. To abort, type any other character.
/R	Read every block on the input device and list the bad sectors but do not perform a transfer unless otherwise specified.
/S	Format the output diskette for single-density use (RX02 drives only).

Table 3-12
DUPLICATE Error Messages

Message	Meaning
CAN'T LOAD INPUT DEVICE	The name of the input device specified in the command line is not a permanent device name.
CAN'T LOAD OUTPUT DEVICE	The name of the output device specified in the command line is not a permanent device name.
COMPARE ERROR	When using the /M option all the areas that do not match are printed as COMPARE ERRORS.
DEVICE IS NOT RX	Either the input or output device specified is not one of the following: RXA0: SYS: RXA1: DSK:
INPUT DEVICE READ ERROR	Bad data on input device; bad sectors.
NO INPUT DEVICE	No input device is specified.
OUTPUT DEVICE READ ERROR	Bad data on output device; bad sectors.
OUTPUT DEVICE WRITE ERROR	Hardware disk-write error.
WRONG FLOPPY TYPE	Attempt to duplicate a double-density diskette between an RX02 drive and an RX01 drive.

3.15 EDIT COMMAND

The EDIT command calls the OS/78 Editor which allows the editing of already existing files. These files must contain ASCII source code.

Format:

```
EDIT outdev:file.ex<indev:file.ex/options
```

In response to the monitor's prompt (.), type the necessary command line and press the RETURN key. The file is now open and the output file is ready for modification. After you press the RETURN key, the number sign (#) is displayed on the terminal, informing you to proceed with an Editor input command. (See Chapter 4 for a detailed explanation of the OS/78 Editor and its options.)

Example:

```
.EDIT RL0B:FILE1.FT<RL1B:TABLE.FT
#
```

In this example, the EDIT command opens output file FILE1.FT on output device RL0B: and opens input file TABLE.FT on input device RL1B:.

NOTE

Since no output device is specified, the default option (DSK:) is used as the output device.

The EDIT command can recall arguments from a previous CREATE or EDIT command. This allows the EDIT command to be used without any arguments. When you restart the system with the START or BOOT button or with the BOOT command, the arguments are lost. However, if you specify both an input and an output file, only the arguments up to but not including the left-angle bracket (<) are remembered.

Example 1:

```
.DA 27-FEB-79
.DA
TUESDAY FEBRUARY 27, 1979
.CREATE MATCH.PA
#A
   .
   .
   .
#E
.EDIT
#R

(MATCH.PA is remembered by
the EDIT command)

     .
     .
     .
#E
```

The above example shows a sequence of commands in creating and editing of file MATCH.PA. The EDIT command is given without any argument since it remembers the file MATCH.PA specified with the CREATE command.

If the date has changed and a command line is typed containing the EDIT command without any arguments, the error message BAD RECOLLECTION will be displayed.

Example 2:

```
.EDIT A2.FT<A1.FT
#R
 .
 .
 .
#E
 .
 .
 .
.EDIT
```

Since the second EDIT command had no arguments, the previous argument up to the left-angle bracket (<) is recalled. Thus, the second EDIT command is equivalent to:

```
.EDIT A2.FT
```

This command executes the CCL.SV and EDIT.SV programs.

EXECUTE

3.16 EXECUTE COMMAND

The EXECUTE command performs the following functions:

1. assemble or compile, link, load and execute a BASIC, FORTRAN, or PAL8 source program
2. link, load and execute an already assembled or compiled program
3. execute an already linked and loaded program

Format:

```
EXECUTE indev:file.ex
```

To use the EXECUTE command with a source file, the input device must be specified. Then that source file is assembled or compiled, linked and loaded into memory and executed. The file extension used determines the compiler or assembler called.

If the file name extension is not a standard extension, specify the correct assembler or compiler by using one of general-purpose dash options in the -ex portion of the command line (see Table 2-3).

Example:

```
.EXE RXA1:NABS.IN-PA
```

If the EXECUTE command is used with an already assembled or compiled program, the input device must be specified. When the EXECUTE command is processed, the binary file is loaded into memory and executed. If the file name extension is not specified in the command line but is specified in the directory, the correct extension is automatically added.

The EXECUTE command can remember arguments from a previous COMPILE, LOAD, PAL, or EXECUTE command. When you restart the system with the START or BOOT button or with the BOOT command, the arguments are lost.

NOTE

For options and expanded command strings acceptable to the EXECUTE command, refer to the appropriate language/loader section.

This command executes CCL.SV and one or more of the following programs: PAL8.SV and ABSLDR.SV, F4.SV, FRTS.SV and LOAD.SV, or BCOMP.SV and BLOAD.SV.

FORMAT

3.17 FORMAT COMMAND

The FORMAT command verifies and formats RL01 cartridge disk packs. You must use FORMAT on each new disk pack before you can store any OS/78 files on it. You should also use FORMAT to reformat any disk pack that has blocks that the system cannot access to make them invisible to the system. (After using FORMAT, you must also use the ZERO command to create a new directory.)

NOTE

FORMAT does not alter the contents of a disk pack that has no bad blocks. If there are bad blocks, the data residing in the good blocks that follow a bad block will become unusable after reformatting, so be sure to make copies of the files that you want to save before using FORMAT.

Format:

```
FORMAT RL01/n/P
```

where:

/n — specifies the drive number, 0 or 1. The default value is 1.

/P — causes the system to pause before and after the operation so you can change disk packs. If you select drive 0, the pause is automatic. Type a CTRL/C to return to the monitor. If you are running under BATCH, FORMAT does not pause after it completes the operation.

Example:

```
.FORMAT RL01
```

This command formats the disk pack on drive 1.

The FORMAT command executes the RLFRMT.SV program.

Table 3-13
FORMAT Error Messages

Message	Meaning
BAD COMMAND LINE	The command line contains a syntax error. Retype the line correctly.
BAD DISK	Disk pack cannot be reformatted. There are more than 63 bad blocks or the system area (blocks 0 - 70 octal) contains bad blocks.
CAUTION - THIS COMMAND DESTROYS CONTENTS OF SYSTEM DISK. CHANGE DISKS BEFORE PROCEEDING OR CTRL/C.	Drive 0 was selected for formatting. Be sure to replace your system disk pack with the one that you want to format.
READY. STRIKE CARRIAGE RETURN TO CONTINUE	The /P option was selected. Mount the disk pack to be formatted and type a carriage return to begin formatting.

GET

3.18 GET COMMAND

The GET command loads memory image files with .SV extensions from an input device back into the same location in memory from which it was saved.

Format:

```
GET indev:file.ex
```

Example:

```
.GET RXA0:JOBCNT
```

If you specify a file name with no extension, an .SV extension is assumed.

Example:

```
.GET RL0B:FILE3
```

In this example, FILE3.SV is loaded into memory from device RL0B.

The specified input file and its Job Status Word are loaded into memory. The entire Core Control Block is sent to the system device where it can be referenced and maintained (see Section 3-31).

The GET command is commonly used to load save files from a device into memory. These save files can be executed either by the START or the EXECUTE command. In addition, you can debug them with octal debugging program (see ODT command).

NOTE

You cannot use the GET command to execute OS/78 system programs.

The monitor performs the GET function.

HELP

3.19 HELP COMMAND

The HELP command accesses the help command's text file called HELP.HL and displays its contents on the terminal. The file contains useful information about the OS/78 commands that you specify as an argument to the command (see Table 3-14). (See Table 3-15 for HELP error commands.)

Format:

```
HELP outdev:file.ex<arg
```

where:

outdev:file.ex	is the output device and file name where the help information is being sent. If no output device is specified, the terminal (TTY) is assumed. If no output file extension is specified the .HL extension is assumed.
arg	is usually an OS/78 command (see Table 3-14).

By typing the HELP command without any arguments, you can obtain a list of all OS/78 commands for which you can get help.

```
.HELP
```

If you want details about a specific command, type the name of the command for which the information is desired, for example:

```
.HELP PAL
```

To obtain a list of all legal arguments for HELP, type the HELP command followed by an asterisk or type the HELP command followed by 'HELP' as follows:

```
.HELP *
```

or

```
.HELP HELP
```

This command runs the HELP.SV and CCL.SV programs.

Table 3-14
OS/78 Help Command Arguments

ASSIGN	LOAD
BASIC	MAP
BOOT	MEMORY
CANCEL	ODT
COMPARE	PAL
COPY	PALERR
CREATE	QUEUE
CREF	R
DATE	RENAME
DEASSIGN	REQUEST
DELETE	RUN
DIRECT	SAVE
DUPLICATE	SET
EDIT	SQUISH
EXECUTE	START
F4ERR	SUBMIT
FORMAT	TERMINATE
GET	TYPE
HELP	UA
LIST	ZERO

Table 3-15
HELP Error Messages

Message	Meaning
DEVICE FULL	The output device becomes full when writing or is already full before output begins.
FETCH ERR	Cannot load handler for the specified device.
NO HELP	There is no file named HELP.HL on the system device.
NO HELP FILE	The input file specified does not exist as a HELP file.
NO TTY HAND	There is no handler in the system for the terminal.
READ ERR	An error occurred while reading the specified HELP file.
WRITE ERR	An error occurred while writing to the output file.

LIST

3.20 LIST COMMAND

The LIST command prints the contents of the specified input files on the line printer (LPT).

Format:

```
LIST indev:file1.ex...file5.ex/options
```

Example:

```
.LI PROD.BA
```

Options for the LIST command are shown in Table 3-16. Error messages are shown in Table 3-17.

When you use the LIST command, the contents of each input file are listed on LPT in the same order the file names are specified in the command line.

Example:

```
.LI RXA1:FL1.FT,FL2.FT,MASFL.FT
```

Although the format specifies that only five files can be named in the command line, more than five files can be listed on the line printer through the use of the wildcard construction.

You can restrict the list of files you want with the command line options. When you specify the /C option, all the specified input files with the current date are listed. If the /O option is used, all specified input files with dates other than the current date are listed. If the /V option is used all files other than those you specified in the command line are listed. If you do not know which files you want, use the /Q option to go through the directory file by file. When you specify the /Q option, each file name followed by a question mark is printed on the terminal so you can decide whether or not you want that file in the list. If yes, type a Y and that file is listed. If no, type any other character and that file is ignored. This continues until all specified input files have been processed.

Example:

```
.LIST*.PA/Q
TOT.PA?Y
MASFIL.PA?N
UPDT06.PA?N
FINTOT.PA?Y
```

As a result, the files named MASFIL.PA and UPDT06.PA are not listed.

This command causes the execution of both the CCL.SV and FOTP.SV programs.

Table 3-16
LIST Options

Option	Meaning
/C	List only those files having the current date.
/O	List only those files having a date other than the current date.
/Q	Each time a file name followed by a question mark is printed on the terminal, type a Y (yes) to list the files or any other character (No) if the file is not to be listed.
/V	List all the files not specified in the command line.

Table 3-17
LIST Error Messages

Message	Meaning
BAD INPUT DIRECTORY	The directory on the specified input device is not valid OS/78 device directory.
BAD OUTPUT DEVICE	A non-file structured device was specified as the output device.
ERROR ON INPUT DEVICE, SKIPPING (file name)	The file specified is not transferred, but any previous or subsequent files are transferred and indicated in the new directory.
ERROR ON OUTPUT DEVICE, SKIPPING (file name)	The file specified is not transferred, but any previous or subsequent files are transferred and indicated in the new directory.
ERROR READING INPUT DIRECTORY	Self-explanatory.
ILLEGAL*	An * was entered as an embedded character in a file name, for example, TMP*.BN.
ILLEGAL?	A ? was entered in an output specification.
NO FILES OF THE FORM xxxx	No files of the form (xxxx) specified were found on the current input device group.

LOAD

3.21 LOAD COMMAND

The LOAD command loads into memory either an absolute PAL binary file (.BN extension) or a relocatable FORTRAN binary file (.RL extension).

When you use the LOAD command without arguments, it will use the arguments from the last COMPILE, LOAD, EXECUTE or PAL command. When you restart the system with the START or BOOT button or with the BOOT command, the arguments are lost.

3.21.1 PAL8 Absolute Binary Files

Absolute binary files are loaded into memory and are ready for execution.

Once a program file is loaded into memory, the following three things can be done.

1. Start the program, using the START command.
2. Save the program, using the SAVE command.
3. Examine or debug the program, using the ODT command.

Format:

```
LOAD indev:file1.ex...file9.ex/options
```

Along with the file specifications for the command line are the options described below.

Example:

```
.LOAD FL3.BN,RUNSEQ.BN
```

NOTE

You can use the ODT command to examine and alter the contents of memory after your files are loaded.

3.21.1.1 **Concatenated Programs (/S option)** - The /S option loads into memory a group of programs that you have concatenated into a single file.

3.21.1.2 **Loading Programs in Specified Areas (/ff and =ffnnnn options)** - Absolute binary files can be loaded into a field other than the one they would normally load into by using /ff option, where ff is an octal number 0 through 37 that represents the field. By specifying the /n option, all specified input files are loaded beginning with the indicated field at a loading address equal to the address specified in the file.

Both the field and the starting address can be specified by using the =ffnnnn option

where:

ff is an octal number 0 through 37 that represents the field

nnnn is a 4-digit octal number 0000 through 7777 that represents a starting address

NOTE

If ffnnnn is all zeros, the default option (field 0 starting address 0200) is used.

3.21.1.3 **Programs not Using Fields 0 and 1 (/8 and /9 options)** - If locations 0-1777 in field 0 are not being used by the program, use the /8 option in the command line. By specifying the /8 option, the contents in these locations are not saved nor is the Command Decoder (system program) reloaded. In effect, these unnecessary operations are not performed.

If locations 0-1777 in field 1 are not being used by the program, use the /9 option in the command line. When you specify the /9 option, the contents in these locations are not saved and the User Service Routine (system program) is not reloaded. In effect, these unnecessary operations are not performed.

3.21.1.4 **Editing or Patching a SAVE File (/I option)** - Files that are already assembled and saved on a device can be changed by using the /I option in the command line. When you specify the /I option, SAVE files are loaded and treated as absolute binary files. This allows you to patch an existing save (.SV) file without reassembling the entire program.

CAUTION

If you want to load more than one file, concatenate the files and use the /S option. The command sequence:

```
.LOAD PROG.SV/I
.LOAD PROG1.SV/I
```

will not work the way you may think it should because part of PROG.SV may be destroyed when CCL.SV system program runs to process the second LOAD command.

3.21.1.5 **Execution After Loading (/G option)** - To execute a program immediately after loading it, use the /G option in the command line. When you specify the /G option, execution takes place immediately after loading is completed. Control is transferred to and remains with the executing program until completion or until the program transfers control back to the monitor.

3.21.1.6 **Clearing Memory After Loading (/R option)** - If you have already loaded a number of files and want to change some or all of the files loaded into memory, use the /R option in the command line. When you specify the /R option, the portion of memory where your files were loaded is reset (or cleared) to appear as though nothing had been loaded.

The LOAD command runs the CCL.SV and ABSLDR.SV programs.

3.21.2 FORTRAN Relocatable Binary Files

Since relocatable FORTRAN binary files are dependent programs, both files and system programs or subroutines are loaded and linked in memory.

Format:

```
LOAD outdev:file.ex<indev:file.ex/options-ex
```

The options summarized below and described in detail in Chapter 7.

After files are loaded, the system library FORLIB.RL is searched for programs or subroutines that are referenced by the program. When the subroutines are found, they are loaded and linked to the program.

NOTE

The COMPILE command with the /L and /G options lets you compile, link, and execute without using the LOAD and EXECUTE commands.

3.21.2.1 **Specifying Additional Input Files (/C option)** - If you have more than nine FORTRAN input files to load, use the /C (continue) option in the command line. When you specify the /C option, the remaining files are accepted as valid input and can be entered on the next command line. Terminate each continuation line (except the last) with a RETURN. Terminate the last line with an ESC.

3.21.2.2 **Including System Symbols in the Symbol Map (/S option)** - If a symbol map output file is specified in the command line with an .RL file and you want the system symbols also included in the map, use the /S option in the command line. The /S option includes all system symbols in the symbol map and marks them with a preceding number sign (#).

3.21.2.3 **Execution After Loading (/G option)** - To execute a program immediately after loading it, use the /G option in the command line. If you terminate the command line with a RETURN, execution takes place immediately after loading is completed. If you terminate the command line with an ESC, the FORTRAN run-time system prints an asterisk (*) to signify that you can enter I/O specifications (see Section 7.1.3). Terminate each assignment line except the last with a RETURN and the last with an ESC. Control is transferred to and remains with the executing program until completion or until the program transfers control back to the monitor.

When relocatable binary files are loaded and linked using the LOAD command, both the CCL.SV and LOAD.SV programs are executed.

MAP

3.22 MAP COMMAND

The MAP command produces a map of all the memory locations used by the specified absolute binary files. The LPT line printer is the default output device for this command. Terminal output may be requested through the use of the -T general-purpose dash option. Options for the MAP command are in Table 3-18; error messages are in Table 3-19.

Format:

```
MAP outdev:file.ex<indev:filel.ex...file9.ex/options-ex
```

MAP command options are described in Table 3-18.

NOTE

If a file name without an extension is specified as the output file, an .MP extension is automatically appended. Input extensions default as described for the LOAD command.

The output of the MAP operation is a series of lines, each of which is comprised of a string of digits. Each digit represents a single memory location, and can have the value 0, 1, 2 or 3. The value means the following:

Value	Meaning
0	The location was not loaded into.
1	The location was loaded into once.
2	The location was loaded into twice.
3	The location was loaded into three or more times.

The appearance of a 2 or 3 may imply a programming error (for example, two separate routines are each trying to load values into the same location).

Each line of digits represents 100 (octal) memory locations. All lines are blocked in groups of two, whereby, each group represents a page. If the resulting map is sent to the terminal, it is bordered on top by a set of octal digits which range from 00-77. Each pair of octal digits corresponds to one memory location (see following example).

It is also bordered down the left hand column by a set of octal digits which are multiples of 64 (decimal). Each set of octal digits corresponds to the first memory location on that line. To determine the address (relative to zero) of a specific location in memory, find the line containing that location. Take the set of octal digits bordering that line and the pair of octal digits that the specific memory location falls under and add both values together.

The following is an example of output on terminal:

```
     BITMAP V4 FIELD 0

       0000000011111111222222223333333344444444555555556666666677777777
       0123456701234567012345670123456701234567012345670123456701234567

00000  2222222221111111100000000000000000000000000000000000000000000000
00100

00200  2222222222222222222222222222222222222222222222222222222222222222
00300  2222222222222222211111111111111111111111111111111111111112222222

00400  2222222222222222222222222222222222222222222222222222222222222222
00500  2222222222222222222222222222222222222222222222222222222222222222

00600  1111111111111111111111111111111111111111111111111111111111111111
00700  1111111111111111111111111111111111111111111111111111111111111111
01000
01000
01100

01200
01300

01400
01500

01600
01700
.
.
.
```

In this example, the location resulting from the addition is 322.

The following is an example of output directed to a line printer:

```
     BITMAP V4 FIELD 0

00000 22222222 21111111 10000000 00000000 00000000 00000000 00000000 00000000
00100

00200 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222
00300 22222222 22222222 22111111 11111111 11111111 11111111 11111111 11122221

00400 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222
00500 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222

00600 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
00700 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

01000
01100

01200
01300

01400
01500

01600
01700
.
.
.
```

If any of the specified input files contain more than one program, only the first binary program in each file is used in the MAP command. The /S option in the command line allows MAP to use all binary programs in the specified input files. If more than one program is being mapped and you want all maps stored in one memory field, use the /n option, where n is an integer (0-37) that represents the field number. When you specify the /n option, all the maps of programs in the specified input file are constructed in the field you specified as n.

If more than nine input files are to be specified in the MAP command, specify only nine input files, then press the ESCape key. Pressing this key causes the Command Decoder to print an asterisk (*), which allows you to continue to specify more input file specifications. Each line, except the last, must be terminated by the RETURN key. If you specify a wrong input file, use the /R option at the end of that line. When you specify the /R option, memory is reset to look as though nothing had been read in. This is shown in the following example. Press the ESCape key to terminate the last line and to cause the command to be executed.

Example:

```
.MAP A,B,C,D,E,F,G,H,I  (ESC)
*J
*K
*L/R  (ESC)
```

If only a terminal is available and no line printer, the style of output on the terminal can be changed to look as though it came from a line printer by using the /T option. Similarly, the style of output on the line printer can be changed to appear as though it came from the terminal.

This command executes the CCL.SV and BITMAP.SV programs.

Table 3-18
MAP Options

Option	Meaning
/n	Confine the construction of all maps to the field you specify as n (n=0-37).
/R	Reset the map just constructed in memory to look as though nothing has been read in.
/S	Instead of mapping only the first binary file contained in the specified input files, map every absolute binary file in the specified input files.
/T	If output is sent to the terminal (default), change the format of the map to that of the line printer and if output is sent to the line printer, change the format of the map to that of the terminal.

Table 3-19
MAP Error Messages

Message	Meaning
BAD CHECKSUM, FILE #n	File number n of the input file list had a checksum error.
BAD INPUT, FILE #n	A physical end of file has been reached before a logical end of file, or extraneous characters have been found in binary file n.
ERROR ON OUTPUT DEVICE	Error occurred while writing on output device.
I/O ERROR FILE #n	An I/O error occurred in input file number n.
NO /I	Cannot produce a map of an image file.
NO INPUT	No binary file was found on the designated device.

MEMORY

3.23 MEMORY COMMAND

The MEMORY command displays the amount of memory that your DECstation has and sets a software limit on the amount of memory OS/78 will use.

Format:

```
MEMORY n
```

where:

n is an octal number representing the number of 4K memory fields available for use by OS/78 in the range: 0-7. If you do not specify a value, the command displays the amount of memory available in the hardware and any software limit that you specified previously up to 32K.

NOTE

Do not change the size of memory while a symbiont is running (see Appendix H).

The following table lists the values of n and their meanings.

n	Memory
0	all available memory
1	8K
2	12K
3	16K
4	20K
5	24K
6	28K
7	32K

Example:

```
.MEMORY 3

16K MEMORY
```

To find the amount of memory actually being used by OS/78, type the command with no argument.

```
.MEM

12K OF 16K MEMORY
```

In this example, a 16K system has been restricted to only 12K of available memory. This was done by using a MEMORY 2 command.

If all available memory is being used, the total amount of memory is printed.

Example:

```
.MEM

16K MEMORY
```

This command executes the CCL.SV program.

ODT

3.24 ODT COMMAND

The ODT command is a debugging tool. It allows you to interpretively control execution and make alterations to a memory-resident program.

Format:

```
ODT
```

The program in memory is available for examination and modification in memory and controlled execution through the use of ODT instruction breakpoints. For more detailed information on ODT, see Chapter 9.

The monitor processes the ODT command.

PAL

3.25 PAL COMMAND

The PAL command assembles a PAL8 source file, producing an absolute binary file (.BN extension) and optional program listing and symbol cross reference listing.

Format:

```
PAL dev:binfile,dev:listfile,dev:creffile<dev:infile/options
```

Example:

```
.PAL ASSEM1.PA
```

This command assembles a source file on SYS called ASSEM1.PA and produces a binary file on SYS called ASSEM1.BN.

If a file name is specified without an extension in the command line, .PA is assumed.

The PAL command can recall arguments from a previous COMPILE, LOAD, or EXECUTE command.

Chapter 5 describes the PAL8 assembler and its command string options.

This command executes the PAL8.SV and CCL.SV programs.

QUEUE

3.26 QUEUE COMMAND

The QUEUE command controls the demonstration symbiont line printer spooler program (SPOOLR) described in Appendix H.

R

3.27 R COMMAND

The R command loads into memory the specified save file from the system device (SYS) and initiates execution.

NOTE

The difference between the R and RUN command is that the R command assumes the program is on device SYS, while the RUN command makes no such assumption. In addition, the RUN command writes on SYS during execution to save the program's Core Control Block for subsequent use in saving and/or starting the programs. The Core Control Block is not transferred to the system device when the R command is executed.

Format:

```
R file.ex
```

Example:

```
.R TEST
```

When a file name is specified without an extension, a .SV extension is assumed.

The R command is most commonly used on programs that are not going to be resaved. Use the RUN or GET commands with programs you want to update (using, for example, ODT) and save.

NOTE

You cannot use the R command to execute OS/78 system programs.

The monitor processes the R command.

RENAME

3.28 RENAME COMMAND

The RENAME command changes the name of the input file to the name specified as the output file. The same device must be specified in both the input and output specifications. Wildcards may be used with this command. Options for the RENAME command are in Table 3-20.

Format:

```
RENAME dev:file.ex<dev:file.ex/options
```

Execution of the RENAME command changes the file name, and a message followed by the old file name is displayed on the terminal.

Example:

```
.REN DSK:FILE.PA<DSK:RECORD.PA

Files Renamed:
RECORD.PA
```

This new file name replaces the old file name in the input directory. The creation date and the contents of the input file remain the same.

Example:

```
.RENAME *.BK<*.FT
```

renames all files with .FT extensions to .BK extensions.

Table 3-20
RENAME Options

Option	Meaning
/C	Rename the input file only if it has the current date.
/O	Rename the input file only if it has a date other than the current date.
/T	After renaming the file, change the date of the new output file to today's date.
/V	Rename all files other than the one specified in the command line.

RENAME executes the CCL.SV and FOTP.SV programs.

REQUEST

3.29 REQUEST COMMAND

The REQUEST command initiates symbiont task operation by setting available memory to 12K words (MEM 2), then loading and starting the specified symbiont. The default extensions for a symbiont save image file are .SM and .SV.

Format:

```
REQUEST dev:symbiont-name.ex
```

This command can be used on DECstation 78 series systems only. Refer to Appendix H for more information.

RUN

3.30 RUN COMMAND

The RUN command loads the specified memory image (.SV) file in memory, transfers its Core Control Block onto the system device, and then initiates execution at the program's starting address. The RUN command is equivalent to a GET command followed by a START command.

Format:

```
RUN indev:file.ex
```

Example:

```
.RUN RXA1:PROG.
```

If a file name is specified without an extension, the extension .SV is assumed.

NOTE

You cannot use the RUN and R commands to execute the OS/78 system programs.

The monitor processes the RUN command.

SAVE

3.31 SAVE COMMAND

The SAVE command saves the portion of memory being used as a memory-image (or save) file. The memory used is indicated by the contents of the Core Control Block saved on SYS in a monitor area or as explicitly specified by you. This new file is given the default extension .SV if you do not specify an extension.

Format:

SAVE input dev:filename.ex ffnnnn-ffmmmm,ffpppp,...;ffssss=cccc

where:

ffnnnn-ffmmmm — ffnnnn is a 6-digit octal number representing the beginning address of a contiguous portion of memory to be saved. The beginning address consists of a field number (ff) in the range 0-37 followed by the memory location within the field (nnnn).

ffmmmm is a 6-digit octal number representing the ending address of the portion in memory. The ending address consists of the same field number as the beginning address (ff) in the range 0-37 followed by the memory location within the field (mmmm).

NOTE

The beginning and ending addresses must be the same field. Crossing field boundaries is not permitted.

ffpppp — ffpppp is a 6-digit octal number representing the address of one location in memory. This represents saving the entire page in which the given location occurs. The address consists of a field number (ff) in the range 0-37 followed by the memory location within the field (pppp).

;ffssss — ;ffssss is a 6-digit octal number preceded by a semicolon that represents the starting address of the program to be saved. The starting address consists of a field number (ff) in the range 0-37 followed by the memory location within the field (ssss).

=cccc — =cccc is a 4-digit octal number preceded by an equal sign that represents the contents of the Job Status Word. The bits in the Job Status Word shown in Table 3-21 control memory allocation for the same file.

Table 3-21
Job Status Word

Bit Condition	Meaning
Bit 0=1	File does not load into locations 0-1777 in field 0, (0000-1777).
Bit 1=1	File does not load into locations 0-1777 in field 1, (10000-11777).
Bit 2=1	Program must be reloaded before it can be restarted because it modifies itself during execution.
Bit 3=1	Program being run will not destroy the BATCH monitor.
Bit 4=1	A memory image file that was generated through the LINKER contains overlays.
Bit 5	Reserved for OS/78 system programs.
Bits 6-9	Unused, and reserved for future expansion.
Bit 10-1	Locations 0-1777 in field 0 need not be saved when calling the Command Decoder overlays.
Bit 11=1	Locations 0-1777 in field 1 need not be saved when calling the USR.

If you use SAVE with no arguments, the necessary information is automatically taken from the current Core Control Block (stored by the last GET, RUN, LOAD or EXECUTE command).

3.31.1 Restrictions on Arguments of the SAVE Command

The arguments used with the SAVE command have several restrictions. They are as follows:

1. The output device must be explicitly specified in the command line. It does not default to DSK.

```
.SAVE DSK:FILE1
```

2. The beginning and ending addresses of an area in memory (fnnnn-fmmmm) must both be located in the same field. Crossing field boundaries is not permitted.

   Example:

```
.SAVE SYS: EXAMPL.BN 20055-20643
```

3. Once an area on a page is specified, that entire page is saved. If another area on the same page is also specified, an error message is sent to the terminal.

   Example:

```
.SAVE RXA1:FL2 10077-10122, FL2 10156-10177
SAVE ERROR
```

4. If a field number is not specified in the starting and ending addresses of an area in memory, field zero is used. Otherwise, the field number must be specified.

   Example:

   ```
   .SAVE SYS: FILE1 0-177,201-377
   ```

5. Saving the area between locations 7600-7777 in any field is permitted. However, locations 7600-7777 in fields 0 and 1 should not be saved because the resident portion of the monitor resides there. Therefore, if the area between 7600-7777 is going to be saved, limit it to fields 2 and 3 and even then only if necessary (due to BATCH).

6. If you specify an address on a page having an odd page number, that page will be saved with the previous page. This is automatically done by the system.

   Example:

   ```
   .SAVE DSK: PROG1 2434
   ```

   In this example, the location is on page 12 which is even. Thus, the contents of page 12 are saved (2400-2577).

   Example:

   ```
   .SAVE DSK: PROG2 2634
   ```

   In this example, the address 2634 is on page 13 which is odd. Therefore, in order to save page 13, page 12 is also saved. This means that the effective area being saved is 2400-2777.

The monitor processes the SAVE command.

SET

3.32 SET COMMAND

The SET command modifies the operating characteristics of OS/78 and its devices, according to options that are specified. SET command device attributes are in Table 3-21. OS/78 device handlers are listed in Table 3-22; error messages are in Table 3-23.

Format:

SET device [[NO]] parameter argument

where:

device	indicates the OS/78 device handler that is to be modified or replaced
NO	is an optional modifier that indicates that the feature specified is to be disabled. NO cannot be used with every attribute
parameter	is the feature to be modified (see Table 3-22 and the following paragraphs.)
argument	is a optional variable (numeric or alphabetic) that is supplied by you

Table 3-22
Summary of SET Command Forms

TTY,SLUx,VLUx	SYS	LPT	HANDLER	Any I/O Device
ARROW COL n ECHO ESC HEIGHT m PAGE *PAUSE n READONLY SCOPE WIDTH = n	INIT xxxxx	COL n LC WIDTH = n	See Section 3.32.1.13	READONLY

3.32.1 SET Command Forms

3.32.1.1 **ARROW** - This parameter causes the handler to map any CTRL character sent to the terminal into the ASCII codes for uparrow (^) and the graphic symbol that is associated with the control code.

* Applies to TTY only.

Format:

SET ter: ARROW

where:

ter: is TTY, SLU2, SLU3, VLU2, or VLU3 (default condition)

For example, the code for CTRL/X (030) would be sent to the terminal as two codes 136 (^) and 130 (X).

To remove this feature, use the NO modifier in the command line. This causes the true control codes to be sent to the terminal.

Example:

```
.SET TTY: NO ARROW
```

3.32.1.2 **COL n** - This parameter changes the number of columns used to print the directory listing (DIRECT command) from two (its default) to the decimal number you specify as n.

Format:

SET TTY COL n

where:

n is a decimal value in the range 0-7.

To remove the value you specified as n and return to its default value of 2, you must specify 2 as the argument n.

Example:

```
.SET TTY COL 2
```

3.32.1.3 **ECHO** - This parameter enables character echoing from the TTY SLUx and VLUx handlers (default system condition).

Format:

SET ter: ECHO

where:

ter: is TTY, SLU2, SLU3, VLU2 or VLU3

To inhibit character echoing at the terminal, use the NO modifier with the command.

Example:

```
.SET TTY: NO ECHO
```

3.32.1.4 **ESC** - This parameter causes the handler to map the code for ESC (033) into the code for the dollar sign ($) character (044) before sending it to the terminal.

Format:

```
SET TTY ESC
```

To remove this feature, use the NO modifier with the ESC attribute.

Example:

```
.SET TTY NO ESC
```

3.32.1.5 **HEIGHT m** - This parameter changes the number of lines (m) that are displayed on the terminal between pauses (see Section 3.29.8). (The default value of m is 24 (decimal).)

Format:

```
SET ter: HEIGHT m
```

where:

ter: is TTY, SLU2, SLU3, VLU2, or VLU3

m is a decimal number in the range 0-255.

3.32.1.6 **INIT cmd** - This parameter causes the system device to execute the command you specify as cmd when you press the START or BOOT button or use the BOOT command. This command name can contain a maximum of five characters excluding a carriage return.

Examples:

```
.SET SYS INIT HELP
```

causes the system to run the HELP.SV program and display the summary of OS/78 commands for which there is help available.

To execute the command contained in the file INIT.CM, type:

```
.SET SYS INIT
```

To inhibit this feature, use the NO modifier in the command line.

```
.SET SYS NO INIT
```

3.32.1.7 **LC** - This parameter causes the line printer (LPT) handler to accept lower-case characters on input.

Format:

```
SET LPT: LC
```

To remove this feature, use the NO modifier in the command line.

Example:

```
.SET LPT: NO LC
```

3.32.1.8 **PAGE** - This parameter enables the CTRL/S and CTRL/Q features described in Chapter 2 (default system condition).

Format:

SET ter: PAGE

where:

ter: is TTY, SLU2, SLU3, VLU2, or VLU3

To inhibit the CTRL/S and CTRL/Q features, use the NO modifier in the command line.

Example:

```
.SET TTY NO PAGE
```

3.32.1.9 **PAUSE** - This parameter changes the pause time between each screenful of data output to the console terminal from 3 seconds (default) to the decimal number you specify as n.

Format:

SET TTY: PAUSE n

where:

n is a decimal value in the range 0-85. If you specify zero or use the NO modifier, no pause takes place.

3.32.1.10 **READONLY** - This parameter causes the device specified to become a read-only device. Therefore, any output sent to this device will cause an error message informing you that the output device is a read-only device.

Format:

SET dev: READO

where:

dev: is any OS/78 I/O device

To remove the READONLY feature, use the NO modifier in the command line.

Example:

```
.SET TTY NO READO
```

NOTE

The READONLY attribute remains in effect until you restart the system with the START or BOOT button or with the BOOT command.

3.32.1.11 **SCOPE** - When an input line of characters is typed on a video terminal and echoed on the screen, this parameter causes the last character echoed on the screen to be erased each time you press the DELETE key.

Format:

SET ter: SCOPE

where:

ter: is TTY, VLU2 or VLU3

When using printing terminals remove this feature with the NO modifier. The system will echo a backslash (\) each time you press the DELETE key. (System defaults: TTY, VLU2 and VLU3 = SCOPE; SLU2 and SLU3 = NO SCOPE.)

Example:

```
.SET TTY NO SCOPE
```

3.32.1.12 **WIDTH = n** - This parameter changes the column width of the terminal or line printer to the number of columns you specify as n.

Format:

SET ter: WIDTH = n

where:

ter: is TTY, SLU2, SLU3, VLU2, VLU3, or LPT

n is a decimal number and a multiple of 8 in the range 001-255, but not 128. The default value is 80 for TTY, VLU2, and VLU3, and 120 for SLU2 and SLU3.

Example:

```
.SET TTY WIDTH = 72
```

3.32.1.13 **HANDLER** - The SET HANDLER command allows you to replace a device handler that resides in the OS/78 Monitor (system head) with another device handler. A device handler is a system program that operates an input/output device. The Monitor calls a device handler via the USR routine (described in Appendix C), when you use a logical device name in an OS/78 command. Each OS/78 handler may control one or two logical devices depending on the device. The number of device handlers in OS/78 is fixed; you can replace them, but you cannot increase or decrease their number.

The OS/78 Monitor is supplied to you with a standard set of device handlers that allows you to use the devices provided with most DECstation configurations. In addition, a complete set of device handlers (see Table 3-23) is distributed to you on the system disk or diskettes that contain the OS/78 system programs. If your DECstation has a device for which there is no handler in the standard set, you can use the SET HANDLER command to replace one of the existing handlers with the one supplied for the device.

The command has two forms. The first form allows you to replace a device handler, and the second form prints a list of the device handlers residing in the system head. You cannot combine these forms in a single command.

Format 1 - Replace device handlers:

```
SET HANDLER old<new
```

where:

old — is the name of a handler currently residing in the system head that you want to replace.

new — is the name of the handler that you want to insert into the system head. The handler must reside on the system device (SYS) as a save file with a .HN extension.

Example:

```
.SET HANDLER LPT<LQP
```

This command causes the line printer handler LPT to be replaced by the letter quality printer handler LQP that resides on SYS as LQP.HN.

Example:

```
.SET HANDLER RX78C<SLU3
```

This command causes the handler for the RXA2 and RXA3 devices used on DECstation 78 systems to be replaced by the SLU3 device handler.

Format 2 - List on the terminal the device handlers currently in the system head:

```
SET HANDLER/L
```

Table 3-23
OS/78 Device Handlers

Handler Name	Logical Device Names	Device
BAT *	BAT	Batch stream handler
LPT	LPT	LA78 and LA180 Printers
LQP *	LQP	Letter Quality Printer
RL0	RL0A, RL0B	RL01 Cartridge Disk Drive 0
RL1	RL1A, RL1B	RL01 Cartridge Disk Drive 1
RLC	RL0C, RL1C	RL01 Cartridge Disk Drive 0/1

(continued on next page)

* Not part of the standard configuration supplied in the system head.

Table 3-23 (Cont.)
OS/78 Device Handlers

Handler Name	Logical Device Names	Device
RX28C *	RXA2, RXA3	RX01 and RX02 Diskette Drives 2 and 3 on DECstation 88 systems
RX78C	RXA2, RXA3	RX01 and RX02 Diskette Drives 2 and 3 on DECstation 78 systems
RXNS	RXA0, RXA1	RX01 and RX02 Diskette Drives 0 and 1 on DECstation 78 and 88 systems
SLU2	SLU2	Serial line handler for hard copy ASCII terminals
SLU3 *	SLU3	Serial line handler for hard copy ASCII terminals
VLU2 *	VLU2	Serial line handler for ASCII video terminals
VLU3 *	VLU3	Serial line handler for ASCII video terminals
VXA0 *	VXA0	Extended Memory Device on DECstation 88/97 systems only

* Not part of the standard configuration supplied in the system head.

Table 3-24
SET Error Messages

Message	Meaning
?CAN'T - DEVICE DOESN'T EXIST	A nonexistent device was referenced.
?CAN'T - DEVICE IS RESIDENT	No modifications are allowed to the system handler.
CAN'T - HANDLER ALREADY RESIDENT	Attempt to replace a handler with one that was already in the system head.
?CAN'T - OBSOLETE HANDLER	The handler has an old version number.

(continued on next page)

Table 3-24 (Cont.)
SET Error Messages

Message	Meaning
CAN'T - TOO MANY LOGICAL DEVICES	The system does not allow you to have more than 12 logical devices (excluding SYS, DSK, and TTY). The handler that you are attempting to insert will cause the number of logical devices to exceed 12.
?CAN'T - UNKNOWN VERSION OF THIS HANDLER	The version of the handler is not one recognized, possibly because it is a newer version.
DONE	The requested operation was successful.
?ILLEGAL WIDTH	A WIDTH argument of 0 or greater than 255 was specified. For the TTY, a width of 128 or one not a multiple of 8 was specified.
INCOMPATIBLE MONITOR - OS/78 V3.0 EXPECTED	The system head contains an old version of the OS/78 Monitor. You cannot change handlers in a system head that has an old version of the monitor.
?I/O ERROR ON SYS:	An I/O error occurred while trying to read or rewrite the handler.
I/O READ ERROR	A hardware error occurred while reading from the system device.
I/O WRITE ERROR	A hardware error occurred while writing to the system device.
?NUMBER TOO BIG	The number specified was out of range.
OLD HANDLER NOT FOUND IN MONITOR	You attempted to replace a handler that is not in the system head.
?SYNTAX ERROR	Incorrect format used in SET command or NO specified when not allowed.
?UNKNOWN ATTRIBUTE FOR DEVICE dev	An illegal attribute was specified for the given device.
xxxxxx.HN NOT FOUND ON SYS	The handler that you want to insert into the system head does not reside on the system device (SYS) or does not have a .HN extension.

SQUISH

3.33 SQUISH COMMAND

The SQUISH command eliminates any embedded empty files on the specified device.

Format:

```
SQUISH outdev:<indev:
```

If you specify an input device only, the operation is performed on that device. If you specify both input and output devices, the files are copied to the output device from the input device. If the input device is a system device, the system head is preserved during the transfer. Before the operation begins, the system asks you to verify that you specified the correct device.

Example:

```
.SQ RXA1:

ARE YOU SURE?Y
```

If yes, type a Y and the files are compressed.

```
.SQ SYS:

ARE YOU SURE?N

.
```

If no, type any other character and the operation is ignored.

Example:

```
.SQ SYS:

ARE YOU SURE?Y
```

In this example, all embedded empty files on the system device are eliminated.

CAUTION

> A device error during a SQUISH operation will leave the entire contents of the device corrupted in a non-obvious way. Therefore, do not use this command unless you have a backup copy of both the system and other files available.

This command executes the CCL.SV and PIP.SV programs.

START

3.34 START COMMAND

The START command initiates execution of the program that is in memory either at the specified starting address in the command line or the starting address specified in the current Core Control Block.

Format:

START ffnnnn

where:

ffnnnn — is a 6-digit octal number representing the starting address of the program. The starting address consists of a field number (ff) in the range 0-37 followed by the memory location (nnnn) within the field. If the starting address is not specified in the command line, the program is started at the address specified in the Core Control Block.

Example:

```
.START 10555
```

In this example, this command line will execute a program starting in field 1 at location 555.

The monitor processes the START command.

SUBMIT

3.35 SUBMIT COMMAND

The SUBMIT command provides batch processing with the option of spooling to output files. Options for the SUBMIT command are listed in Table 3-24.

Format:

```
SUBMIT spooldev:<batchfile.ex/options
```

Example:

```
.SU RLOB:BTCHIN
```

If no input file extension is specified, .BI is assumed.

Along with the necessary arguments for the command line are SUBMIT options described in Table 3-24.

Table 3-25
SUBMIT Options

Option	Meaning
/E	Treat all errors as non-fatal errors so batch processing continues uninterrupted.
/H	Process the batch input file without echoing and without sending the $JOB and $END batch commands to both the terminal and batch log.
/Q	Send only the $JOB and $MSG batch commands, and all results from batch processing to the output device.
/T	Send all output from batch processing to the terminal only.
/U	Do not interrupt batch processing and ignore anything typed at the keyboard with the exception of CTRL/C.

The SUBMIT command is commonly used to run multiple programs and sequences of system commands that require little or no interaction with the user or operator.

3.35.1 Processing and Terminating a Batch Input File

The SUBMIT command starts processing your batch input file by printing a job header. The job header is printed on the LA78 line printer if it is available. (The LQP78 line printer is not supported in batch processing.) If an LA78 line printer is not available, the terminal is used as the output device to print a log of batch operations. Processing continues with the OS/78 commands and BATCH commands following the $JOB batch command and terminates with the $END batch command. Termination can also occur from an error in the OS/78

commands. However, batch processing is allowed to continue in spite of errors if the /E option is used in the command line. When you specify the /E option, all errors are treated as non-fatal errors and batch processing continues without interruption.

If the ESCape key is used to terminate an OS/78 command in the batch file, it will cause your batch run to terminate unless the /E option is specified. To terminate all the commands in a batch file, use the RETURN key. If CTRL/C is typed during batch processing, the batch run is terminated, and control returns to the monitor.

If you want to log the $JOB and $MSG commands, and batch results on the output device, and disregard the OS/78 commands and comment lines, use the /Q option. If you have an LA78 and want all batch output sent to the terminal instead, use the /T option in the command line. When you specify the /T option, batch output is sent to the terminal only.

If you want your batch input file to be processed without interruption, use the /U option in the command line. If you specify the /U option, any errors are non-fatal, and your batch input file is processed without interruption. Anything you type on the terminal except CTRL/C during batch processing is ignored.

If you want your batch input file to be processed without echoing its contents on the terminal and without sending the $JOB and $END batch monitor commands to the terminal and BATCH log, use the /H option. When you specify the /H option, echoing, $JOB, and $END batch monitor commands are suppressed. Statements that are not suppressed by the /H option are as follows:

1. Any OS/78 command messages and user program messages sent to BATCH
2. User program messages sent to the terminal
3. Batch run-time error messages
4. $MSG commands

NOTE

Do not use the /Q and /H options in the same command line.

3.35.2 Spooling

The SUBMIT command also provides optional spooling of output files. Spooling is a process whereby output intended for unit-record devices (such as a terminal or line printer) is redirected to a file on disk or diskette. This provides a convenient way for producing multiple copies of a listing. It also allows an installation that has a slow printer to perform computations rapidly and to review the results later when processing time is not at a premium.

To use the spooling option during batch processing, specify an output device, the system disk as the input device and the batch file in the command line. The output device specified as the spool device must be a file-structured device (disk or diskette). If no output device is specified, spooling does not take place.

Example:

```
.SU RXA1:<RXA0:BATCHX
```

NOTE

Your batch input file must be on the system device (SYS).

When spooling is specified, output ordinarily directed to a non-file structured device is intercepted and stored in temporary files on the spool device. The system automatically assigns a name to each file. The first file is called BTCHA1. The second is called BTCHA2 and so on up to and including BTCHA9. If you have more than nine input files, the names continue with BTCHB0 up to and including BTCHZ9.

See Chapter 8 for a detailed explanation of OS/78 Batch.

This command executes the CCL.SV and BATCH.SV programs.

TERMINATE

3.36 TERMINATE COMMAND

The TERMINATE command causes the system to terminate OS/78 operation and load and execute a user-written program. The program must reside on SYS in save file (memory image) format and have the name TERMIN.SV. You must press the START or BOOT button to resume OS/78 operation after using this command.

Format:

```
TER
```

This command executes the CCL.SV program.

TYPE

3.37 TYPE COMMAND

The TYPE command displays the contents of the specified input files on the terminal screen. Options for the TYPE command are in Table 3-25; error messages are in Table 3-26.

Format:

```
TYPE ter:<indev:filel.ex...file5.ex/options
```

where:

ter: is TTY, SLU2, SLU3, VLU2, or VLU3. (default is TTY)

indev:filel.ex is the device and file specification. You can enter up to five different specifications, each separated by a comma.

Along with the necessary arguments for the command line are the TYPE options described below and summarized in Table 3-26.

Table 3-26
TYPE Options

Option	Meaning
/C	Type those files with the current date.
/O	Type those files with a date other than the current date.
/Q	Each time a file name followed by a question mark is printed on the terminal, type a Y (yes) to type the file or any other character (no) if the file is not to be typed.
/V	Type all files that were not specified in the command line.

The TYPE command displays the contents of each input file on the terminal in the same order the file names are specified in the command line.

Example:

```
.TY SYS:MASFL
```

Although the format specifies that only five file names can be used in the command line, more than five files can be typed on the terminal by using wildcards.

3.37.1 Using TYPE Options

Use the /C or /O option if the file's date is to determine what is to be displayed. When you specify the /C option, all the specified input files with the current date are typed. If the /O option is used, all specified input files with dates other than the current date are typed. The /V option displays all files other than the ones that were specified. The /Q option displays each file name followed by a question mark. If you want to display that file, type a Y. If not, type any other character and that file is ignored. This continues until all specified input files have been processed.

NOTE

Use the LIST command to print files on the LA78 line printer (LPT).

This command executes the CCL.SV and FOTP.SV programs.

Table 3-27
TYPE Error Messages

Message	Meaning
BAD INPUT DIRECTORY	The directory on the specified input device is not valid OS/78 device directory.
BAD OUTPUT DEVICE	A non-file structured device is inappropriately specified as the output device.
ERROR ON INPUT DEVICE, SKIPPING (file name)	The file specified is not tranferred, but any previous or subsequent files are transferred and indicated in the new directory.
ERROR ON OUTPUT DEVICE, SKIPPING (file name)	The file specified is not tranferred, but any previous or subsequent files are transferred and indicated in the new directory.
ERROR READING INPUT DIRECTORY	Self-explanatory.
ILLEGAL*	The wildcard character * was entered as an embedded character in a file name, e.g., TMP*.BN.
ILLEGAL?	The wildcard character ? was entered in an output specification.
NO FILES OF THE FORM xxxx	No files of the form (xxxx) specified were found on the current input device group.

UA, UB, UC

3.38 UA, UB, UC COMMANDS

The UA, UB, and UC commands are three separate commands that save their specified arguments in a temporary area. Only a single OS/78 command can be used as an argument. These three commands can recall their arguments when the command that saved the OS/78 command is typed without any arguments. Once recalled, the OS/78 command is executed. When you restart the system with the START or BOOT button or with the BOOT command, the arguments are lost.

Format:

```
UA
UB  } any OS/78 command line
UC
```

Type the command followed by the command line you want to save, then press RETURN. This saves the specified command line in a temporary area. There are three areas, one for each command. To recall and execute the stored command, type the command that saved it, but do not specify any arguments.

Example:

```
.UA COPY RXA1:<DSK:RECALL
```

In this example, the system stores the COPY command and its arguments in a temporary file.

Example:

```
.UA
```

In this example the OS/78 command previously specified with the UA command and stored in a temporary file is recalled and executed.

These commands are most commonly used with OS/78 commands that are repeated throughout a batch file. The use of these instructions decreases the amount of input required.

The monitor processes the UA, UB, and UC commands.

ZERO

3.39 ZERO COMMAND

CAUTION

Use this command with extreme care, since it can erase the contents of a system device, including the system head.

The ZERO command clears the directory on a device and deletes all files stored on that device. Use it only for devices that contain your programs and data files. If used on a system device, it will destroy the system head and convert it to a files-only device.

Format:

ZERO dev:

The command executes the CCL.SV and PIP.SV programs.

Example:

```
.ZERO RXA1:
```

When you use the DIRECT command on the device just zeroed, a blank directory will appear showing all blocks are available.

If the specified device is SYS, the system asks you if this is the correct device. If yes, type a Y and the directory is zeroed. If no, type any other character, and the operation is ignored. Remember that if you type a Y, the system on the specified device will be destroyed.

CHAPTER 4

THE SYMBOLIC EDITOR

4.1 INTRODUCTION

The Editor allows you to create and modify ASCII text files. These files may contain assembly language source programs, FORTRAN and BASIC source programs, or any other information that has the format of character strings.

The Editor is a helpful tool; however, it must be told precisely what to do. You direct its operation by typing commands in the form of a single letter or a letter with arguments and, in most cases, pressing the RETURN key directly after the command line.

This chapter describes the procedures you follow to create a file and the commands you use to modify it.

4.2 CALLING THE EDITOR

The CREATE and EDIT commands call and run the Editor.

The Editor operates in two modes: command mode and text mode. In command mode, the Editor displays a number sign as a prompt (#) and interprets the characters you type on the terminal as instructions to do some job on the text in its buffer. In text mode, the Editor interprets the characters you type as text. The two modes -- and the controls you use to request them -- are described in Section 4.3.

4.2.1 Creating a New File - The CREATE Command

The CREATE command summons the Editor to let you open and write a new file. The format is

```
CREATE outdev:file
```

CREATE accepts no input specifications and only one file name and device for output. You provide the input by typing in text at the terminal.

After you press the RETURN key to execute the command, the system Editor displays a number sign (#) on the screen to indicate that it is ready to receive your first instruction.

Thus,

```
.CREATE RXA1:RUN1.PA
#
```

opens a file named RUN1.PA on output device RXA1.

Before you can type in text from the terminal, you must put the Editor into text mode. For details on text mode, see Section 4.3.1.

4.2.2 Editing an Existing File - The EDIT Command

The EDIT command summons the OS/8 Editor to let you retrieve and work on a source program previously stored as a file. The format is

```
EDIT outdev:file=indev:file1,...indev:file9
```

The Editor signals with a number sign (#) as soon as it is ready to accept your first instruction.

To work on a source program that you have created and stored as a file (or sequence of files), enter the file or files as input in the EDIT command line. EDIT will accept up to nine input files in a line.

The Editor allows only one output file in a command line. You must specify an output file to receive the modified version of your source program.

For example, the following command opens input file TABLE.FT on RXA0 and a file called FILE1.FT on RXA1 for output (The Editor signals when ready.):

```
.EDIT RXA1:FILE1.FT<RXA0:TABLE.FT
#
```

To cause the Editor to read in the first page of the input file, type R in response to the number sign. (For details on Editor commands, see Section 4.3.2.)

4.3 MODES OF OPERATION

The OS/8 Editor operates in two modes: the command mode and the text mode.

In the command mode, the Editor prints a # on the terminal to indicate that it is waiting for you to type a command on the keyboard.

In text mode, the Editor accepts anything you type at the keyboard as part of the file you are creating or modifying.

The key commands in Table 4-1 enable you to transfer between modes or return control to the monitor.

Table 4-1
Editor Key Control Commands

Command	Mode in Which Used	Meaning
CTRL/C	Text and Command Modes	Returns control to the monitor. All text that has been edited is lost. CTRL/C should be used with utmost caution, since no output file will be stored.
CTRL/O	Command Mode	Stops the listing of text. Returns control to Command Mode.
CTRL/L	Text Mode	Returns the Editor to Command Mode.

4.3.1 Text Mode

To put the Editor in text mode so that you can enter a new file -- or modify or add to one that you have already created -- type the Insert or Append command.

Format:

```
I RETURN
```

or

```
A RETURN
```

These commands cause the Editor to place the text that you enter at the terminal into its text buffer. If you use the Insert command, the Editor stores the text before the first line of any existing material in the buffer. The Append command instructs the Editor to place the text you enter after the last line of existing text in the buffer.

The Editor accepts text in both upper and lower case.

To enter a line of text that you have typed on the terminal, press the RETURN key.

For example:

```
*I
HEAD OF THE BUFFER
```

or

```
*A
BOTTOM OF THE BUFFER
```

Before you type RETURN, read the line over for errors. Make corrections with the DELETE key or the CTRL/U key command. DELETE erases the last character you typed. CTRL/U deletes the entire line. (CTRL/U is equivalent to typing DELETE back to the beginning of the line.)

To correct a line that you have sent to the buffer with RETURN, you must put the Editor in command mode with CTRL/L and use the appropriate editing commands (see Section 4.3.2.4).

The buffer holds approximately 5600 characters (decimal). When 256 locations remain, the Editor rings the warning bell on the terminal. From this point until the buffer is full, typing RETURN causes the Editor to enter a line of text, then switch to command mode and ring the terminal bell. You may continue to enter text by this method one line at a time until the Editor detects the absolute end of its buffer.

To continue, you must first empty the buffer. The Page command enables you to send the contents of the buffer -- or any part of it -- to an output device. To use the Page command, return to command mode with CTRL/L.

Format:

```
P
```

or

```
nP
```

or

```
m,nP
```

where:

n	is a line you want to send to an output device.
n,m	is a sequence of lines (n through m) that you wish to send to an output device

The P command automatically appends a form feed to the output, thus producing a page of text. This allows you to paginate the contents of your file.

Before you start typing in the next page, make sure that no text remains in the buffer. To do this, use the Kill command (see Section 4.3.2.3), which clears the buffer. Then type the Append command (to put the Editor back in text mode) and continue entering your source program.

To return to command mode at any point, type CTRL/L.

To end the session -- that is, to place all remaining text in the output file, close the file, and return control to the monitor -- use the Exit command.

Format:

```
E (RET)
```

4.3.2 Command Mode

In command mode, the Editor performs the operations you specify on the text in the buffer.

For example, to enter text into the buffer from your input device, use the Read command.

Format:

R (RET)

The Read command instructs the Editor to read a page of text from an input device into the buffer -- that is, to read text until it encounters a form feed character. If the buffer contains text already, the Editor adds the new page to it.

The Editor provides five types of command: Input, Listing, Output, Editing, and Search.

Each command consists of a single letter, preceded optionally by one or two numeric arguments. The letter indicates the operation; the arguments in most cases tell the Editor which lines to act upon.

Enter the commands after the number sign prompt in upper case only.

General format:

X (RET)

or

nX (RET)

or

m,nX (RET)

where:

X	is a command
m,n	are line numbers (m must be less than n.)

Except for noted exceptions, you terminate the command with the RETURN key.

4.3.2.1 **Input Commands** - Input commands (Table 4-2) instruct the Editor to accept text from the terminal (text that you type in) or from an input device (text that you have stored as a file). To execute the commands, type RETURN key.

Special characters, including lower-case letters may be input to the file. The ESCape character is echoed as a dollar sign ($) for readability.

In these commands, the Editor ignores ASCII codes 340 through 376. These codes include the codes for the lower-case alphabet (ASCII 341-372). The Editor returns to the command mode only after encountering a form feed or when the text buffer becomes full.

Table 4-2
Editor Input Commands

Command	Meaning
A	Append the text being typed at the keyboard until a form feed (ASCII 214 or CTRL/L) is encountered. The form feed returns control to command mode. Text input following the A command is appended to whatever is currently in the text buffer.
I	Insert whatever text is typed before line 1 of the text buffer. The form feed (CTRL/L) terminates the insertion process and returns control to the command mode.
nI	Insert whatever text is typed (until a form feed is typed) before line n of the text buffer.
R	Read one page from the input device specified to the EDIT or CREATE commands, and append the new text to the current contents of the buffer. If no input file was indicated or if no input remains, a question mark (?) is printed and the Editor returns to the command mode.

4.3.2.2 **Listing Commands** - Listing commands (Table 4-3) display on the terminal all or part of the contents of the text buffer. Type RETURN key to execute.

Table 4-3
Editor Listing Commands

Command	Meaning
L	List entire contents of the text buffer on the terminal.
nL	List line n of the text buffer on the terminal.
m,nL	List lines m through n of the text buffer on the terminal.
G	Get and list the next line that has a label associated with it. A label in this context is any line of text that does not begin with one of the following: space (ASCII 240) / (ASCII 257) TAB (ASCII 211) RETURN (ASCII 215) At the termination of a G command, control returns to the command mode with the current line counter equal to the line just listed.

(continued on next page)

Table 4-3 (Cont.)
Editor Listing Commands

Command	Meaning
nG	Get and list the first line that begins with a label, starting the search at line n.
B	Print the number of available memory locations in the text buffer. The Editor returns the number of locations on the next line. To estimate the number of characters that can be accommodated in this area, multiply the number of free locations by 1.7.

The Editor remains in command mode after a list command and updates the value of the current line counter to be equal to the number of the last line printed.

4.3.2.3 **Output Commands** - Output commands (Table 4-4) send text from the buffer to a device you specify for output. Type RETURN key to execute.

Table 4-4
Editor Output Commands

Command	Meaning
E	Output the current buffer and transfer all remaining pages of input to the output file; close the output file and enter it in the directory. When this buffer is full, the text is output to the indicated output file. The E command automatically outputs a form feed after the last line of output, and returns control to the monitor. NOTE If you do not use the E command to close a file after editing, any changes, additions, or corrections will not appear in the output file. Thus, the E command should usually be the last command that you enter in an editing session (also see Q command).
P	Write the entire text buffer to the output file.
nP	Write line n of the text buffer to the output file.

(continued on next page)

Table 4-4 (Cont.)
Editor Output Commands

Command	Meaning
m,nP	Write lines m through n, inclusive, to the output file. NOTE The P command automatically appends a form feed to its output, thus producing a page of text. This command allows you to paginate your listing. However, if the K command is not used after a P command, the text remains in the buffer and is again output with the new text read in before the next P command.
K	Kill the buffer. All text is deleted from the text buffer. NOTE The Editor ignores the commands nK or m,nK, with the result that you cannot destroy the buffer by mistyping a List command (m, nL).
Q	Immediate end-of-file. The Q command causes the text buffer to be output. The file is then closed (entered into the directory with the current date as its creation date), and control returns to the monitor.
N	Write the current buffer to the indicated output file and read the next logical page. The N command is equivalent to a P, K, R command sequence.
nN	Write the current buffer to the output file, kill the buffer, and read the next logical page. This is done n times until the nth logical page is in the text buffer. Control then returns to command mode. (The N command cannot be used with an empty text buffer, since there is no text to be written. If the buffer is empty when the N command is attempted, a question mark (?) is printed.) For example, to read in the fourth page of a file, give the commands #R (to read the first page) and #3N (to read three more pages)

(continued on next page)

Table 4-4 (Cont.)
Editor Output Commands

Command	Meaning
V	The V command causes the entire text buffer to be listed on the line printer. The V command only works with the LA78 line printer. It does not work with the LQP78 line printer.
nV	List line n of the buffer on the line printer.
m,nV	List lines m through n, inclusive, on the line printer.

During an editing session, the device you specify for output may become full before you have written out the entire file.

If this happens and you attempt to write more text on the device, an error occurs. The Editor immediately closes the file and prints the following message and prompt.

```
FULL
*
```

To continue, you must specify a new output device and file name. Since the Editor retains the contents of its buffer, no text is lost.

When you have finished the editing job, combine your output files.

4.3.2.4 **Editing Commands** - Editing commands (Table 4-5) permit deletion or alteration of text in the buffer. Press the RETURN key to execute.

Table 4-5
Editing Commands: Deletion and Alteration

Command	Meaning
nC	Change the text of line n to the line(s) typed after the command is entered (typing a form feed terminates the text input). The C command is equivalent to a D command followed by an I command.
m,nC	Delete lines m through n, and replace with the text line(s) typed after the command is entered. (Typing CTRL/L indicates the end of the changed lines.) The C command utilizes the text collector in altering text.
nD	Delete line n from the buffer.
m,nD	Delete lines m through n from the buffer.

(continued on next page)

Table 4-5 (Cont.)
Editing Commands: Deletion and Alteration

Command	Meaning
nY	Yank (read) in n pages from the input file into the text buffer, without writing any output. For example, #5Y reads through four logical pages of input, deleting them without producing output. The fifth page is read into the text buffer, and control automatically returns to command mode. NOTE Use this command with caution; it irrevocably deletes the contents of the text buffer.
m,n$pM	Move lines m through n directly before line p in the text buffer. The $ character means that you type the dollar sign key, not ESCape, ALTMODE, or other possibilities. The old occurrence of the moved text is then removed. This command can move one line, but it needs three arguments. You can provide three arguments by specifying the same line number twice. For example, #6,6$21M moves line 6 in front of line 21.

4.3.2.5 **Search Commands** - Search commands (Table 4-6) cause the Editor to search a text for occurrences of characters and strings that you specify. The Editor sets the current line pointer at the line containing the characters you want to find.

Search commands are discussed in detail in Section 4.4.

Table 4-6
Editor Search Commands

Command	Meaning
S	Perform a character search (Section 4.4.1).
J	Perform an interbuffer search for character strings (Section 4.4.2.2).
F	Look for next occurrence of the string currently being sought.
ESC($)	Perform an intrabuffer character string search.

4.3.2.6 **Special Command Mode Characters** - Special characters recognized by the Editor in the command mode are listed in Table 4-7.

Table 4-7
Editor Special Characters: Command Mode

Character	Function
Period (.)	The Editor assigns an implicit decimal number to the line on which it is currently operating. At any given time the period, which represents this decimal number, may be used as an argument to a command. In the following example, the L command is used since it allows text to be listed. Typing #.L means list the current line. Typing #.-1,.+1L means list the line preceding the current line, the current line, and the line following it, and then update the current line counter to the decimal number of the last line printed. The Editor updates the current line counter, represented by the period, as follows: • After an R (Read page) or A (Append) command, the period is equal to the number of the last line in the buffer. • After an I (Insert) or C (Change) command, the period is equal to the number of the last line entered. • After an L (List) or S (Search) command, the period is equal to the number of the last line listed. • After a D (Delete) command, the period is equal to the number of the line immediately after the deletion. • After a K (Kill) command, the period is equal to 0. • After a G (Get and list) command, the period is equal to the number of the line displayed by the G. • After an M (Move) command, the period is not updated and remains whatever it was before the command.

(continued on next page)

Table 4-7 (Cont.)
Editor Special Characters: Command Mode

Character	Function
Slash (/)	The symbol slash (/) has a value equal to the decimal number of the last line in the buffer. It may also be used as an argument to a command. For example, #10,/L means list from line 10 to the end of the buffer.
LINE FEED Key	When the Editor is in command mode, pressing the LINE FEED key has the same effect as .+1L which causes the Editor to display the line following the current one and to increment the value of the current line counter (period symbol) by one. LINE FEED does not perform this function while in the text mode.
Right-Angle Bracket (>)	Typing the right-angle bracket (>) while in command mode is equivalent to typing .+1L and causes the Editor to echo > and then display the line following the current line. The value of the current line counter is increased by one so that it refers to the last line displayed.
Left-Angle Bracket (<)	In command mode, typing the left-angle bracket (<) is equivalent to typing .-1L and causes the Editor to echo < and then print the line preceding the current line. The value of the current line counter is decreased by one so that it refers to the last line printed.
Equal Sign (=)	In the command mode, using the equal sign in conjunction with either the line indicator period (.) or slash (/) causes the Editor to display the decimal value of the argument preceding it. You can find by this method the number of the current line (.=nnnn) or the total number of lines in the buffer (/=nnnn).

(continued on next page)

Table 4-7 (Cont.)
Editor Special Characters: Command Mode

Character	Function
Colon (:)	The colon performs exactly the same function as the equal sign (=).
ESCape Key	When the Editor is in command mode, pressing the ESCape key signals an intrabuffer character search. It echoes as a dollar sign ($) on the terminal screen. When the Editor is in text mode, the Escape key echoes as a dollar sign, but it is stored in the file as an ESCape character (033).

4.4 SEARCHING A TEXT

The following search commands enable you to make additions and corrections in your text. The Editor searches for occurrences of the single character or character string that you specify.

4.4.1 Single-Character Search -- the S Command

The format of a single-character search is:

```
S
x
```

where:

x is the alphanumeric character you want to search for.

To specify a line or a sequence of lines that you want to search, use the following format:

```
nS
c
```

or

```
n,mS
c
```

where:

n is a line number

n,m are line numbers indicating a segment of text

c is any single character you wish to search for

For example, the following command causes the Editor to search lines 20-40 for an occurrence of the character B:

```
#20, 40S
B
```

The Editor displays the character it is searching for and everything preceding it in the line. At this point you can perform the following operations.

- Delete the entire portion of the line not yet displayed and terminate the line and the search by pressing the RETURN key.
- Delete characters from right to left by typing the DELETE key.
- Insert characters after the last one printed simply by typing them.
- Insert a carriage return/line feed, thus dividing the line into two, by pressing the LINE FEED key followed by CTRL/L.
- Continue searching the line to the next occurrence of the search character by typing CTRL/L.
- Change the search character in the line and continue searching by typing CTRL/G(BELL) followed by the new search character. This allows all editing to be done in one pass.
- Type CTRL/G(BELL) twice to terminate the command.

The usual form of the character search command is #.S, followed by the RETURN key and the character to be located. Use this form of the command to modify the current line.

4.4.2 The Character String Search

The Editor can search the buffer for any unique combination of characters. In a character string search, the Editor sets the current line pointer at the line containing the first occurrence of the string.

Two types of character string search are available: intrabuffer and interbuffer.

4.4.2.1 Intrabuffer String Search - In an intrabuffer search, the Editor scans the text in the buffer for the string you specify. If it fails to find an occurrence of the string, it prints a question mark and returns to command mode.

To initiate an intrabuffer search, type the ESCape key in response to the Editor's prompt and enter the string. (ESCape echoes as a dollar sign.) The string must occur in one line.

If you wish to begin the search at line 1 of the buffer, terminate the string with a single quotation mark ('). If you wish to begin the search at the current line + 1, use a double quotation mark (") to terminate the string.

The format of an intrabuffer search command is:

```
$string'
```

or

```
$string"
```

where:

$	is the character the Editor echoes when you type ESCape
string	is a group of up to 20 ASCII characters
' (single quote)	causes the Editor to begin searching at line 1 of the buffer
" (double quote)	causes the Editor to begin searching at current line +1

NOTE

Do not include single or double quotation marks in a string because the Editor recognizes them as instructions.

The Editor places the number of the first line containing the search string in the current line indicator and displays the prompt sign (#). To display the number on the terminal, type a period (.) followed by an equal sign. The format is:

```
.=
```

You can use a line number you obtain this way as an argument in any Editor command.

For example, the following command causes the Editor to search for the first occurrence of the string CDF10, beginning at line 1 of the buffer:

```
#$CDF10'
#
```

The response to this command is revealed as line 35:

```
#.=35
```

Command lines can include more than one instruction. For example, assume that the buffer contains the following text:

```
ABC DEF GJO
1A2B3C4D5E6
.STRINGABCD
.
.
```

To list the line that contains ABC, type

```
#$ABC'L
```

The search begins with line 1 and continues until the Editor finds the string. The Editor sets the current line counter equal to the line in which the string ABC occurred. The L (List) command causes the line to be printed as follows:

```
ABC DEF GJO
```

The Editor returns to command mode, awaiting further commands. If you want to find the next reference to ABC, type:

```
#"L
```

In this case, the quotation marks (") cause the last string the Editor searched for to be used again, with the search beginning at the current line +1. It is not necessary to enter the search string again. The command may be used several times in succession. For example, if you want to find the fourth occurrence of a string containing the characters FEWMET, type

```
#$FEWMET' """L
```

This command will list the line which contains the fourth occurrence of that string. The L command (or any other command code) can follow either ' or ". The L command causes the line to be listed if the Editor finds the string.

To clear the text string buffer, type

```
#$'
```

The Editor responds with a question mark and clears the text string buffer.

The properties of the commands ' and " allow for easy and useful editing, as the following example illustrates. To change the CIF 20 to CIF 10, enter the following commands:

```
#$DUM, '$CIF 20"C
CIF 10      /NEW FIELD
```

The above set of instructions first causes the Editor to start at line 1 and search for the line beginning with DUM,. Then it searches for CIF 20, starting from the line after the line containing DUM,. The line number of the line containing the string CIF 20 becomes the current line number. The C command applies the instructions of the command line to what is typed in the next line -- that is, the string CIF 10.

Since this search feature produces a line number as a result, any operations which require a line number will accept a string instead. For example:

```
#$STRING'+4L
```

lists the fourth line after the first occurrence of the text STRING in the text buffer.

```
#$LABEL1,',$LABEL2,"L
```

lists all lines between the two labels, inclusive.

```
#$PFLUG'S
```

performs a character search on the line which contains PFLUG. (Type the search character after typing the RETURN key that enters the line.)

In commands that include both strings and explicit numbers, strings should appear first. For example, the following commands:

```
#1+$BAD!'L
```

will not list the next line after the string BAD! occurs. The correct syntax is:

```
#$BAD!'+1L
```

4.4.2.2 **Interbuffer String Search -- J Command** - In an interbuffer search, the Editor scans the contents of the text buffer for the character string you specify. If it fails to find an occurrence of the string, it sends the buffer to an output file, clears the buffer, and reads in the next page of text from the input file. The Editor then resumes the search at line 1 of the new buffer. When the input file is exhausted, the Editor prints the number sign prompt (#) and awaits your next instruction.

If the search is successful, the Editor sets the current line indicator equal to the number of the line containing the first occurrence of the string.

The format for an interbuffer search is:

```
J
$string'
```

where:

$	is a prompt character printed by the Editor
string	is a group of up to 20 ASCII characters
' (single quote)	causes the Editor to begin searching at line 1 of the buffer.

To display the number of the line containing the string, type a period (.) after the Editor's number sign prompt (#), followed by an equal sign. The format is:

```
.=
```

For example, the following command instructs the Editor to make an interbuffer search for the string WRITE, beginning at line 1 of the current buffer. The .= construction reveals that line 4 of the current buffer contains the string.

```
#J
$WRITE'
#.=0004
```

To find further occurrences of the string WRITE, type the F command. The F command searches the buffer for the last character string entered, starting from the current line count + 1. The displayed line following the F command line contains a number prompt sign (#), the format you type to obtain a line number (.=), and the line number. The result is:

```
#F

#.=0008
```

This example causes a search for the string WRITE, starting at the current line + 1. If you have specified no output file, the J or F command reads the next input buffer without attempting to produce any output.

NOTE

Use the J command for interbuffer searches only. After the J or F command has processed the entire input file, execute either an E or Q command to close the output file.

Table 4-8 lists the commands used to abort the string search command.

Table 4-8
Aborting Editor String Search Commands

Command	Explanation
CTRL/U	A CTRL/U will return control to the Editor command mode if you type it while entering text in a string search command.
DELETE	Pressing the DELETE key while entering text for a string search causes the text so far entered to be ignored and allows a new string to be inserted. The Editor displays a dollar sign ($) in response.

4.5 A SAMPLE EDITING JOB

You will often find it necessary to edit a closed file to change the code, correct overlooked errors, or insert additional information.

This section shows you how to use editing commands to edit an existing assembly language program. The example program, shown below, is written in PAL8 assembly language. When executed it displays HELLO!. Note that it contains deliberate errors.

```
          *200
          MONADR=7600
START,    CLA CLL           /CLEAR ACCUMULATOR AND LINK
          TLS               /CLEAR TERMINAL FLAG
          TAD BUFADR        /SET UP POINTER
          DCA PNTR          /FOR GETTING CHARACTERS
NEXT,     TFF               /SKIP IF TERMINAL FLAG SET
          JMP .-1           /NO: CHECK AGAIN
          TAD I PNTR        /GET A CHARACTER
          TLS               /PRINT A CHARACTER
          ISZ PNTR          /DONE YET?
          CLA CLL           /CLEAR ACCUMYLATOR AND LINK
          TAD I PNTR        /GET ANOTHER CHARACTER
          SZA CLA           /JUMP ON ZERO AND CLEAR
          JMP I MON         /RETURN TO MONITOR
          JMP NEXT          /GET READY TO PRINT ANOTHER
BUFADR    BUFF              /BUFFER ADDRESS
PNTR,     BUFF              /POINTER
BUFF,     215;212;"H;"E;"L;"L;"O;"!;0
MON,      MONADR            /MONITOR ENTRY POINT
```

You have checked the above program and want to make the following changes:

- Insert a comment line at beginning of program.
- Correct TFF to TSF in line with label NEXT.
- Correct the spelling of accumulator where it appears the second time.
- Transpose lines JMP I MON and JMP NEXT.
- Add a comma after label BUFADR.

To summon the Editor and open the file, type EDIT and the file name, SAMPLE/PA. The Editor will prompt with the number sign (#) to indicate that it is ready to receive the editing commands:

```
.EDIT SAMPLE.PA
#
```

If you are changing a part of a program but want to save your original file, type the following commands:

```
.RENAME SAMPLE.BK<SAMPLE.PA
.EDIT SAMPLE.PA<SAMPLE.BK
```

The Editor will assign the name SAMPLE.PA to the modified version. SAMPLE.BK will remain unchanged.

At this point, the text buffer is empty. Type the R (Read) command to read in the first page of the file, which in the case of this example, is the entire program. The Editor prompts with the number sign (#) when it has executed the command. Use the L command to display the program.

First you want to insert a comment line at the top of the program to serve as a title. To do this, type the I (Insert) command, followed by your text.

```
#I
/ROUTINE TO TYPE A MESSAGE
```

This command inserts whatever text you type before line 1 of the text buffer. The RETURN key is typed after the line to allow the Editor to recognize this text as a separate line. Now type CTRL/L to return to the Editor command mode. Verify that this is now line 1 by typing the command

```
#1L
```

which causes the Editor to display line 1 of the buffer.

```
/ROUTINE TO TYPE A MESSAGE
```

Your next job is to change TFF to TSF.

Use the string search ESCape command to find and display the line

```
#$TFF'L
```

Use the single character search option and the current line pointer to make the correction in the line by typing.

```
*.S
```

followed by the RETURN key and the search character, F.

The Editor will display the line up to the F. Delete the F by pressing the DELETE key and type in the correct character S. Since this completes all changes in this line, type CTRL/G (BELL) followed by another CTRL/G (BELL) to return to command mode.

Now change ACCUMYLATOR to ACCUMULATOR. This example will illustrate the Editor's ability to accomplish repetitive searches for identical character strings. The intrabuffer search commands are shown here to illustrate their use.

The intrabuffer search uses the ESCape key and the single quote (') and double quote (") constructions.

Typing

```
*$CLA'"L
```

(where $ is the echo for ESCape) will list the line containing the second occurrence of CLA, that is,

```
CLA CLL   /CLEAR ACCUMYLATOR AND LINK
```

To correct the word, use the single character search command S with the period construction to indicate the current line.

Type

```
*.S
```

Now type the RETURN key and Y, the character that is the object of the search. Use the DELETE key to erase this character and type the correct character U.

This concludes the corrections on this line. To return to the Editor command mode type CTRL/G twice. Note that if you typed a carriage return after making the correction, control would return to the command mode but the remainder of the line after the corrected character to the end of the line would be deleted.

Verify correction by typing

```
*.L
```

to list the corrected line

```
CLA CLL   /CLEAR ACCUMULATOR AND LINK
```

The next correction transposes the two lines JMP I MON and JMP NEXT.

Use the character string search command J, which will return the number of the line to be moved. Type

```
*J
$JMP I MON'
*.=0016
```

Use the M (Move) command to transpose the line by typing

```
16,16$18M
```

This command moves line 16 (JMP I MOV) before line 18, or after line 17 (JMP NEXT), which in effect performs the transposition. It also deletes the initial occurrence of the moved line, now making JMP I MON the 17th line. Type 17L to verify this.

Next add a comma after the label BUFADR. The G (Get) command allows you to get the next line that has a label associated with it after the line that is equal to the current line counter. Thus, typing

```
#G
```

will get the line starting with BUFADR and display it on the terminal as follows:

```
BUFADR    BUFF    /BUFFER ADDRESS
```

Then make the correction by adding a Comma after the label, using the change or character search commands previously described.

When you have completed all the changes, display the corrected file on the terminal. Type

```
#L
```

and the entire contents of the text buffer will appear.

Remember to close out the file and return to the monitor by typing the E command.

4.6 EDITOR OPTIONS

The Editor provides the following options:

- /B The Editor converts two or more spaces to a TAB when reading from an input device.
- /D The Editor deletes the old copy of the output file (if one exists) before opening the new output file on the device. If you do not specify /D, the Editor does not delete the old copy of the output file until you have transferred all data to the new file with the E or Q command.

4.7 EDITOR ERROR MESSAGES

Two types of error messages, nonfatal and fatal, are generated when an error is made while running the Editor.

Nonfatal errors, such as an incorrect format in a command string or a search for nonexistent information, cause the Editor to display a question mark. For example, if a command requires two arguments and only one is provided, the Editor will display a question mark (?), perform a carriage return/line feed, and ignore the command as typed. Similarly, if you type an illegal or unrecognized command character, the error message ? will be displayed, followed by a carriage return/line feed, the command will be ignored. However, if you

provide an argument for a command that does not require one, the argument may be ignored and the normal function of the command performed. Table 4-9 lists nonfatal errors that you may encounter while using the Editor.

Table 4-9
Nonfatal Editor Error Messages

Condition/Message	Explanation
L ?	The buffer is empty. Nonexistent information is requested.
7,5L ?	The arguments are in the wrong order. The Editor cannot list backward.
17$10M ?	This command requires two arguments before the $; only one was provided.
H ?	Nonexistent command letter.

Major errors (Table 4-10) cause control to return to the monitor. These errors cause a message to be printed in the form:

```
?n ^C
```

where:

n — is an error code listed in the table

^C — indicates that control has passed to the monitor.

These errors generally result in complete loss of the output file.

Table 4-10
Major Editor Error Codes

Error Code	Meaning
0	Editor failed in reading a device. Error occurred in device handler; most likely a hardware malfunction.
1	Editor failed in writing onto a device; generally a hardware malfunction.
2	File close error occurred. For some reason the output file could not be closed; the file does not exist on that device.
3	File open error occurred. This error occurs if the output device is a read-only device or if no output file name is specified on a file-oriented output device.
4	Device handler error occurred. The Editor could not load the device handler for the specified device. This error should not normally occur.

4.8 SUMMARY OF EDITOR COMMANDS AND SPECIAL CHARACTERS

The command and special characters discussed in this chapter are summarized in Table 4-11.

Table 4-11
Editor Command and Special Characters

Command	Format	Meaning
A	A	Append the following text being typed at the keyboard until a CTRL/L (form feed) is typed. The form feed returns control to the command mode. Text input following the A command is appended to whatever is present in the text buffer.
B	B	List the number of available memory locations in the text buffer. The Editor returns the number of locations on the next line. To estimate the number of characters that can be accommodated in this area, multiply the number of free locations by 1.7.
C	nC	Change the text of line n to the line(s) typed after the command is entered. (Typing a CTRL/L terminates the input.)
	m,nC	Delete lines m through n and replace with the text line(s) typed after the command is entered. (Typing CTRL/L indicates the end of the inserted lines.)
D	nD	Delete line n from the buffer.
	m,nD	Delete lines m through n from the buffer.
E	E	Output the text buffer and transfer all remaining pages of the input file to the output file, closing the output file and returning to the monitor.
F	F	Follows a string search. Look for next occurrence of the string currently being sought (by the J command).

NOTE

If the search fails while you are using the F command, further commands cause the system to prompt with a ?. The file must be closed and then reopened.

(continued on next page)

Table 4-11 (Cont.)
Editor Command and Special Characters

Command	Format	Meaning
G	G	Get and list the next line that has a label associated with it. A label in this context is any line of text that does not begin with one of the following: space (ASCII 240) / (ASCII 257) TAB (ASCII 211) RETURN (ASCII 215) At the termination of a G command, control goes to the command mode with the current line indicator (.) equal to the line just listed.
	nG	Get and list the first line that begins with a label, starting the search at line n.
I	I	Insert whatever text is typed before line 1 of the text buffer. (Typing CTRL/L terminates the entering process and returns control to the Editor command mode.)
	nI	Insert whatever text is typed (until a CTRL/L is typed) before line n of the text buffer.
J	J	Interbuffer search command for character strings (see Section 4.4.2.2 describing the InterBuffer Character String Search).
		NOTE If the search fails while you are using the J command, further commands cause the system to prompt with a ?. The file must be closed and then reopened.
K	K	Kill the buffer. Delete all text from the text buffer.
		NOTE The Editor ignores the commands nK and m,nK with the result that you cannot destroy the buffer by mistyping a List command (m,nL).

(continued on next page)

Table 4-11 (Cont.)
Editor Command and Special Characters

Command	Format	Meaning
L	L	List entire contents of the text buffer on the terminal.
	nL	List line n of the text buffer on the terminal.
	m,nL	List lines m through n of the text buffer on the terminal. Control then returns to command mode.
M	m,n$pM	Move lines m through n directly before line p in the text buffer. The $ character represents typing the dollar sign key, and not other possible keys. The old occurrence of the moved text is removed.
N	N	Write the current buffer to the output file and read the next page.
	nN	Write the current buffer to the output file, kill the buffer, and read the next page. This action is repeated n times until the nth page is in the text buffer. Control then returns to command mode. You may not use the N command with an empty text buffer. A question mark (?) is printed if you attempt to do this.
P	P	Write the entire text buffer to the output file. The P command automatically outputs a FORM character (214) after the last line of output.
	nP	Write line n of the text buffer to the output file and a FORM character.
	m,nP	Write lines m through n, inclusive, to the output file and a FORM character.
Q	Q	Immediate end-of-file. Q causes the text buffer to be output and the file closed.
R	R	Read one page from the input device and append the new text to the current contents of the text buffer. If no input file was indicated or if no input remains, a question mark (?) is displayed and control returns to the command mode.
S	S	Character search command (see Section 4.4.1).

(continued on next page)

Table 4-11 (Cont.)
Editor Command and Special Characters

Command	Format	Meaning
V	V	List the entire text buffer on the line printer.
	nV	List line n of the text buffer on the line printer.
	m,nV	List lines m through n, inclusive, on the line printer.
Y	nY	Yank (read) in a logical page from the input file, without writing any output. For example, #5Y reads through four logical pages of input, deleting them without producing output. The fifth page is read into the text buffer, and control automatically returns to the command mode.
$(ESC)	$TEXT' "	Perform a character string search for the string TEXT. Following a string search, #" causes a search for the next occurrence of the string (see Section 4.4.2.1 describing the Intrabuffer Character String Search).
.= or .: /= or /:		Typing these characters obtains the current line number (.=) and the last line number (/=) in the text buffer. The number is printed by the Editor immediately after you type the equal sign. (The colon character is equivalent to the equal sign.)
>	>	Equivalent to .+1L, list the next line in the text buffer.
<	<	Equivalent to .-1L, list the preceding line in the text buffer.
LINE FEED Key		Equivalent to .+1L, list the next line in the text buffer.
#	#	Print the current Editor version number.