Operator's Reference Guide

PREFACE

Your computer is one of the most flexible, versatile, and powerful desktop computers in the world. From now on, you will get your work done faster and more easily than ever before. And you'll produce accurate, attractive results.

This Guide shows you how to use your computer. The information is presented in the order that you need it. Procedures are explained one at a time. Each new term is printed in bold type the first time it's used, and then precise definitions appear in the Glossary.

The introductory Section -- Basic Procedures -- includes information you need to begin using you computer for office work. You will learn how to set up your computer, how to load an operating system, how to work with diskettes, and how to use the basic keyboard (other keyboards are illustrated in Appendix C).

The next Section -- The MS-DOS Operating System -- describes the MS-DOS system structure, including the file system and special editing functions, the diskette management utilities available to you, the internal commands, and EDLIN (the MS-DOS text editor). Also included is a description of the CP/M-Emulator, which runs CP/M programs under MS-DOS.

The third Section -- The CP/M-86 Operating System -- explains CP/M-86 capabilities and features. You will learn this system's components, input/output devices, operating principles, commands, and its file system. This Section also discusses basic operating procedures, device and media management, and ED (the CP/M-86 text editor).

The fourth Section -- Hard Disk Utilities -- applies to you if your machine has the internal or external hard disk option. You will learn how to prepare the hard disk for use with the AUTOSET utility, how to display drive information with the VDIR utility, and how to back-up your hard disk files with the ARCHIVE utility. Also included is a chapter on error messages.

The fifth Section is a brief discussion on how to solve any problems you may encounter. Most problems can be solved easily and quickly; and, for your convenience, a list of error messages is included. You will find this Section a handy reference tool.

The sixth Section consists of useful Appendixes. Appendix A lists display driver specifications, including the 132-column utility. Appendixes B and C, respectively, illustrate the many character sets and keyboards available with your computer (NOTE: The **Programmer's Tool Kit, Volume I,** contains utilities for generating both custom character sets and keyboards.) Finally, Appendix D discusses some of the software development tools available to you -- including the **Programmer's Tool Kit,** the **Graphics Tool Kit,** and **Communications.**

The final Section -- the Glossary -- consists of precise and clear definitions of important terms and concepts.

WARNING: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a CLASS A computing device pursuant to subpart J of part 15 of FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

CONTENTS

SECTION I: BASIC PROCEDURES

SECTION II: THE MS-DOS OPERATING SYSTEM

SECTION III: THE CP/M-86 OPERATING SYSTEM

SECTION IV: HARD DISK UTILITIES

SECTION V: SOLVING PROBLEMS

SECTION VI: APPENDIXES

SECTION VII: GLOSSARY

FIGURES

TABLES

SECTION I

BASIC PROCEDURES

1. SETTING UP YOUR COMPUTER

This chapter describes how to install, plug in, and turn on your computer. If you have not yet unpacked your computer, look inside its box for unpacking instructions.

1.1 PARTS OF YOUR COMPUTER

The parts of your computer are collectively called **hardware.** After you read this guide, you will know how to use every part.

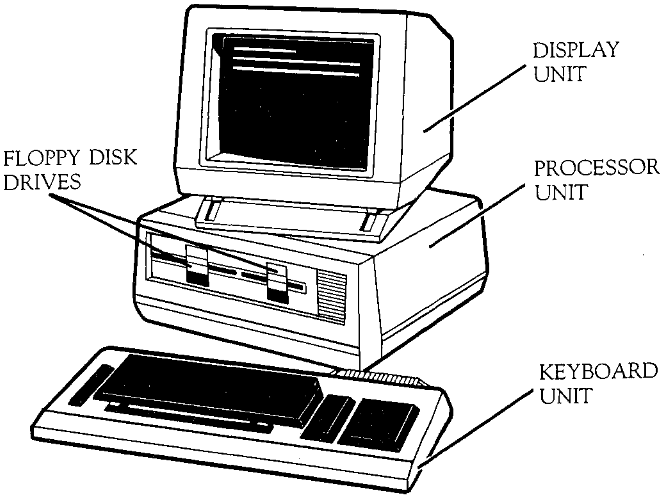
Your computer is available in several different models. Each model contains three basic parts (Figure 1-1):

- The **keyboard.**

- The **processor unit.**

- The **display unit** (also called the **CRT**).

Figure 1-1: Computer with Two Floppy Disk Drives



In addition, your computer may have an internal or external **hard disk** (Figures 1-2 and 1-3).

Figure 1-2: Computer with Internal Hard Disk

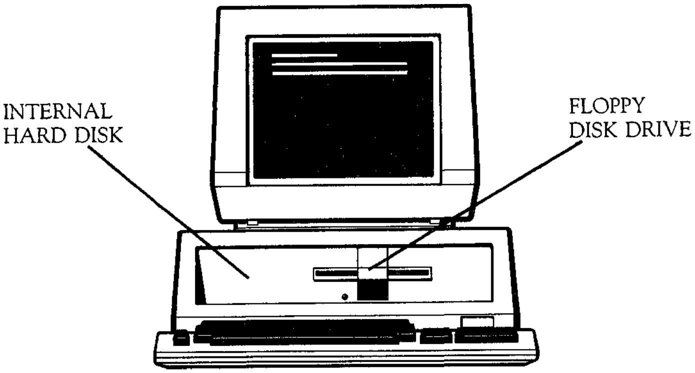
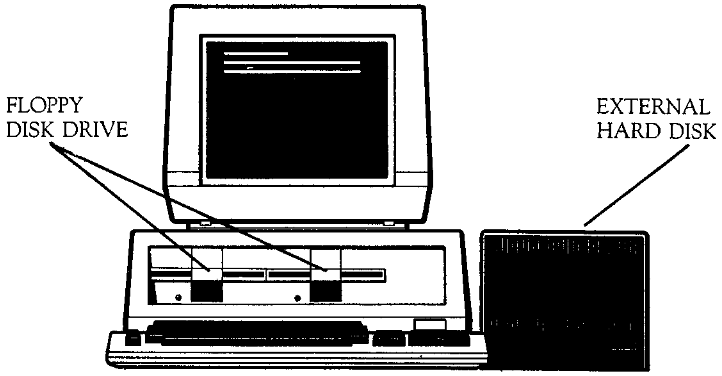


Figure 1-3: Computer with External Hard Disk



The processor unit for your model may have one or two **floppy disk drives.** Computers with an internal hard disk have only one floppy drive; computers with an external hard disk may have one or two floppy drives.

Floppy disk drives can be **single-sided** or **double-sided:**

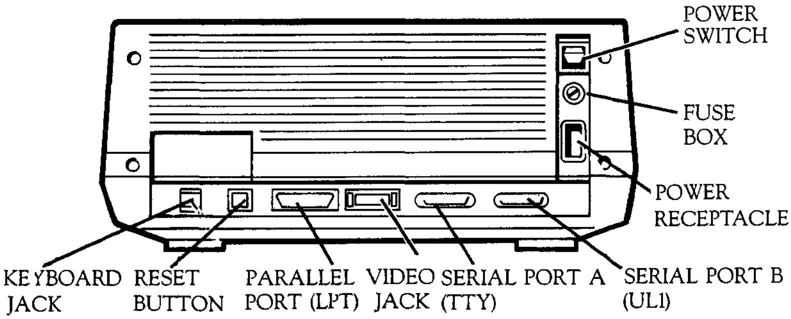
- Single-sided and double-sided drives look alike.

- Computers with double-sided drives are labeled on the front of the processor unit.

- Computers with an internal hard disk always have a double-sided drive.

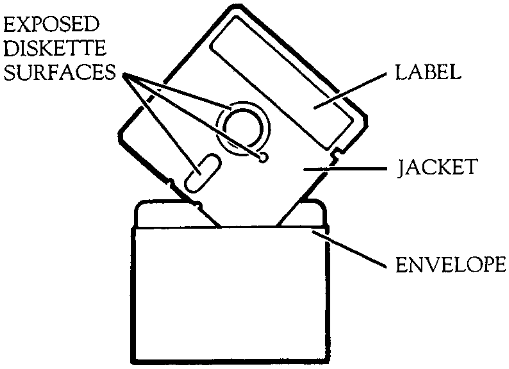
The back of the processor unit contains the power switch, power jacks, and other connectors and switches (Figure 1-4).

Figure 1-4: Processor Unit, Rear View



Another important part of the computer is the **floppy diskette.** Floppy diskettes and the hard disk are used to store the work that you do with the computer. Diskettes are called "floppy" because they are flexible -- as opposed to the hard disk, which is rigid. Figure 1-5 shows the parts of a diskette.

Figure 1-5: Parts of a Diskette



Application programs are distributed on two types of floppy diskettes: single-sided and double-sided.

- Double-sided diskettes are used only with computers that have double-sided floppy disk drives.
- Single-sided diskettes may be used with computers that have single- or double-sided floppy disk drives.

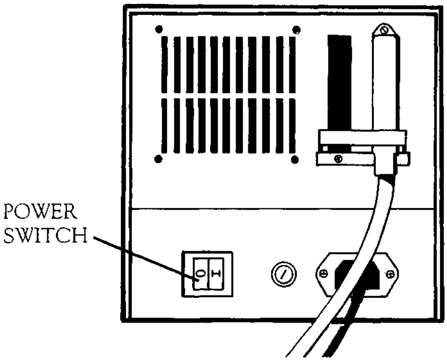
1.2 INSTALLING YOUR COMPUTER

CAUTION: Handle your computer with care. Be careful not to jar the hard disk or allow it to vibrate.

The following steps describe how to install and plug in your computer.

1. Turn off the main power switch (the Ø position; Figure 1-4). If you have an external hard disk, turn off the hard disk power switch (Figure 1-6).

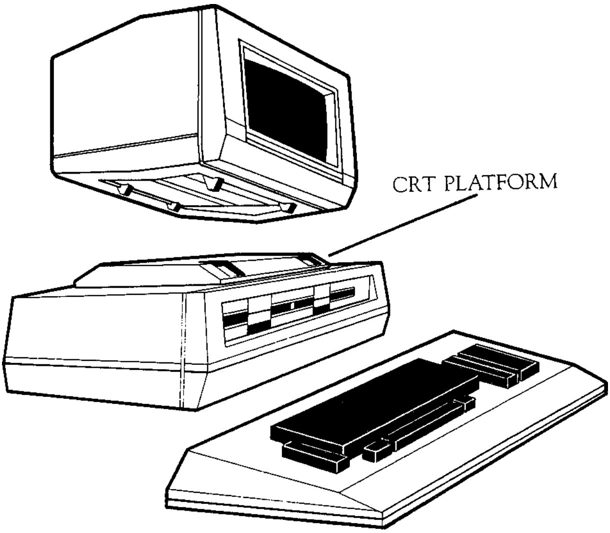
Figure 1-6: External Hard Disk Power Switch



2. Place the CRT platform on the processor unit (Figure 1-7). The hole in the platform bottom fits over the peg in the middle of the processor.

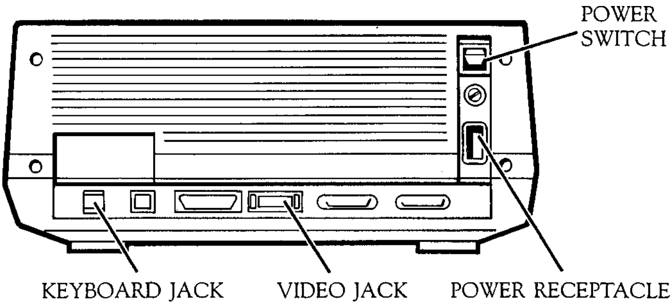
3. Place the CRT in the grooves on the platform.

Figure 1-7: Positioning Computer Units



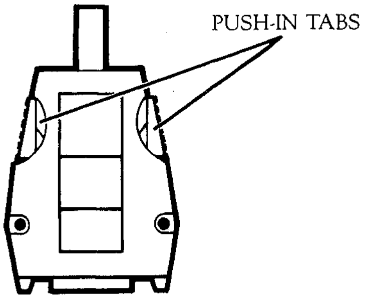
4. Plug the CRT cord into the video jack (Figure 1-8).

Figure 1-8: Processor Unit, Rear View



To disconnect the plug, press in the tabs on each side of the plug while pulling it out (Figure 1-9).

Figure 1-9: Display Unit Connector



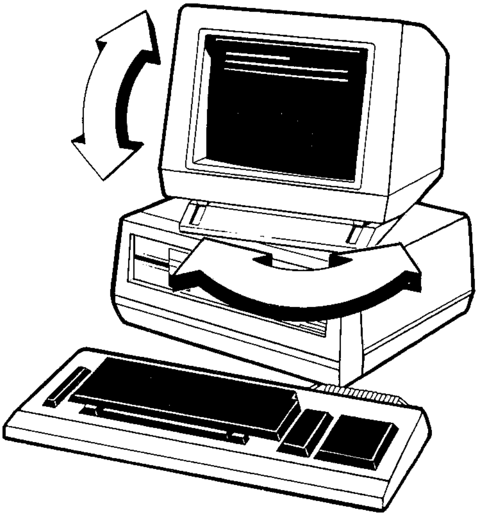
5. Plug the keyboard cord into the keyboard jack (Figure 1-8). The cord can go around or under the processor unit.

6. Plug the power cord into the power socket (Figure 1-8) and into a three-pronged outlet. Always use the computer power cord; other cords cannot be used safely.

7. Adjust the CRT for easy viewing. Push it gently sideways and up and down (Figure 1-1Ø).

8. Place the keyboard anywhere within cord range. Your dealer can supply you with a longer keyboard cord if you need one.

Figure 1-10: Positioning the Display Unit



1.3 TURNING ON THE POWER

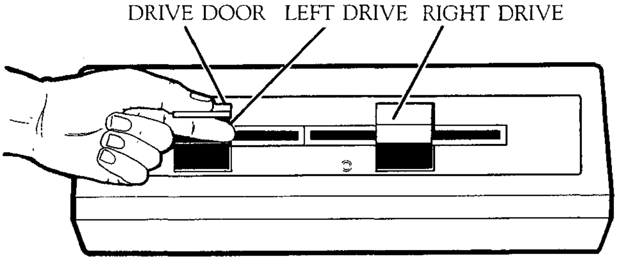
To turn on the power:

1. Open the floppy disk drive or drives:

   - A floppy disk drive is a slot on the front of the processor unit (Figure 1-1). Your computer may have one or two floppy drives.

   - Open the drives by gently lifting the drive doors (Figure 1-11).

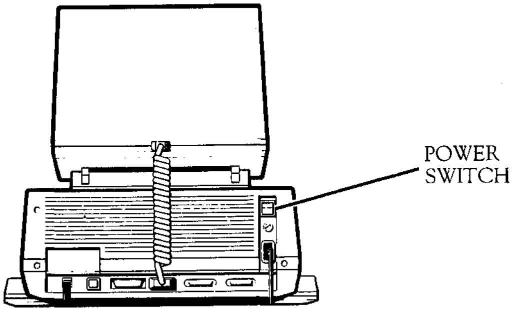
Figure 1-11: Opening a Drive Door



2. Make sure the drives are empty. Remove any cardboard packing material or floppy diskettes.

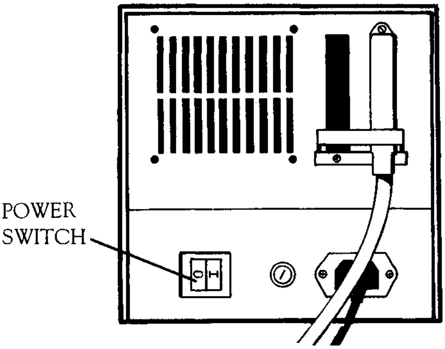
3. Turn on the main power switch (the 1 position; Figure 1-12).

Figure 1-12: Power Switch



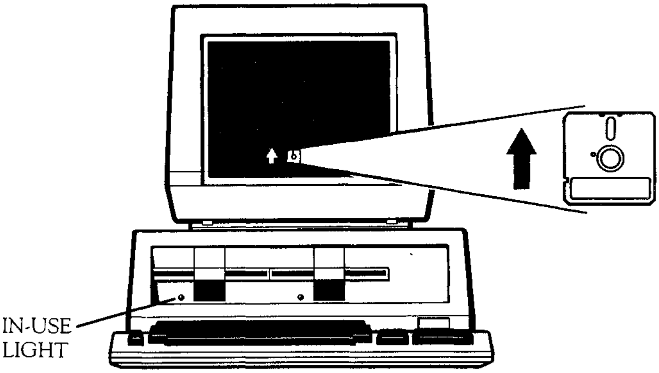
4. If you have an external hard disk, turn on its power switch (Figure 1-13).

Figure 1-13: External Hard Disk Power Switch



When you turn on the power, the computer displays its **power-on message** (Figure 1-14). This tells you to insert a diskette into the computer. Also, the red **in-use light** by the drive lights up, telling you to insert the diskette in the drive. Chapter 2 describes how to insert a diskette.

Figure 1-14: Power-On Display



If you have a hard disk, the computer may load the operating system when you turn on the power, rather than display the power-on message. Loading the operating system is described in Chapter 2.

1.4 TURNING OFF THE POWER

When you have finished using the computer:

1. Check that the red in-use light by each drive has turned off. The light must be **off** before you remove a diskette from the drive.

2. Remove all diskettes from the drives and store them safely.

3. Turn off the power switch for the external hard disk, if you have one.

4. Turn off the main power switch.

You do not need to turn off the power unless you are leaving your computer for a long time. But you should always remove and store your diskettes when you have finished using the computer.

2. USING THE OPERATING SYSTEM

The **disk operating system** (the **DOS**) tells the computer how to do basic functions -- for example, what to display on the screen. The computer cannot function without the DOS.

The DOS consists of a list of instructions to the computer. These instructions are stored on the hard disk or on a floppy diskette. A floppy diskette that contains the DOS is called a **system diskette.**

Your computer can use any of several operating systems. The primary operating system supplied with your computer is MS-DOS. In some cases, CP/M-86 is also supplied.

NOTE: CP/M-86 is not supplied for computers with a hard disk. A CP/M-86 emulator is supplied to allow you to use CP/M-86 with the hard disk (see 8.4).

Each time you turn on the power, the computer transfers the DOS from the hard disk or system diskette into the processor unit. This is called **loading** or **booting** the system.

This chapter describes how to load your computer with MS-DOS. Some details differ if you use CP/M-86.

2.1 LOADING THE OPERATING SYSTEM

You load the DOS with your system diskette. If your system includes a hard disk, you can also set up the hard disk to load the DOS automatically for you. You must use the AUTOSET and SYSCOPY computer programs to do this. (Chapter 8 describes how to use SYSCOPY; Chapter 17 describes AUTOSET.)

To load the DOS with your system diskette:

1. Make sure the power-on message is shown on the screen (see Figure 1-12).

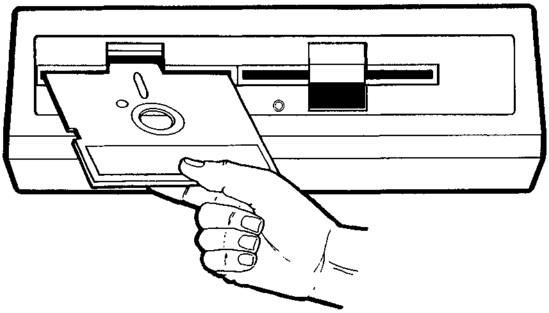
2. Remove the system diskette from its envelope. Hold the diskette by its label.

CAUTION: Handle diskettes carefully.

- o Do not bend diskettes.
- o Do not touch the exposed surfaces.
- o Keep diskettes away from magnetic fields, such as those generated by magnetic paper clip holders or ringing telephones.

3. Insert the diskette -- label side up and label edge last --into the left drive (Figure 2-1). Do not force the diskette. If it sticks, remove it and try again.

Figure 2-1: Inserting a Diskette



4. Gently but firmly close the drive door. DO NOT reopen the drive door until the DOS is loaded.

CAUTION: Handle floppy disk drives carefully:

- Do not stick anything other than a diskette into a drive.
- Open and close drive doors carefully.
- Never open a drive door when its in-use light is on or the drive motor is running.

The DOS loads in about 15 seconds. During that time, the **memory test message** appears. When the DOS is loaded, it shows its **sign-on message**. The following sections describe these messages.

2.1.1 MEMORY TEST MESSAGE

The memory test message shows a clock symbol, the letter "M", and a number.

The clock symbol asks you to wait. The letter M tells you the computer is testing its **memory**. The number indicates memory size.

Memory is information storage. It is measured in **bytes**; one byte contains one character of information, such as a letter or number. The number shown in the memory test message indicates memory size as follows:

NUMBER	MEMORY SIZE
2ØØØ	128K bytes (1K bytes is about 1ØØØ bytes)
4ØØØ	256K bytes
8ØØØ	512K bytes
EØØØ	896K bytes

The letter M and the number appear so briefly you may not notice them the first time the DOS is loaded.

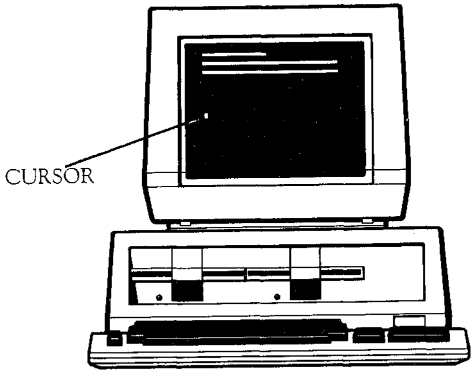
2.1.2 SIGN-ON MESSAGE

The **sign-on message** appears when the DOS is loaded (Figure 2-2). The sign-on message gives setup information about the DOS.

A request that you type the date is shown with the sign-on message. The **cursor** appears beside the request. The cursor shows you where your next typed entry will appear. This time and date information is specific to MS-DOS.

CP/M-86 has a time-of-day facility, but does not stamp the date and time on files or request time or date information upon booting.

Figure 2-2: Sign-On Message



2.2 SETTING THE DATE AND TIME

The DOS keeps track of the date and time while you are using the computer. The DOS uses this information to date your work.

The DOS asks you to set the date and time after it is loaded. Always set the date and time correctly so that the DOS can date your work accurately. In this way you will always know when you did what with the computer.

The DOS asks you to type the date:

- Do not type the name of the day when you change the date.
- You can type the date in European format (day-month-year) by separating the numbers with hyphens.
- You can type the date in American format (month/day/year) by separating the numbers with slashes.
- Press the Return key when done. The Return key is labeled RETURN or ↵
- Don't worry about making mistakes:
  - Use the Backspace key to erase mistakes. The Backspace key is labeled BACKSPACE or ←
  - If you make a mistake, the DOS asks you to retype the date.

The DOS then shows the time it is now set for and asks you to change the time.

NOTE:

- The DOS keeps a 24 hour clock. One o'clock pm, for example, is 13:ØØ; 11 pm is 23:ØØ.
- When the DOS shows the time, it shows the hour, minute, second -- and half second.

To change the time:

- Do not type the half second.
- Separate the time with colons (hour:minute:second). The second colon and number for the second is optional.
- Press the Return key when done.
- Correct mistakes as when setting the date.
- If you don't want to change the time, press Return when asked to type the new time.

Once you have set the date and time, you can check (and change) them any time you are using the DOS:

- To check the date, type "date" and press Return.
- To check the time, type "time" and press Return.
- The DOS shows the date or time and asks you to reset it. Press Return to leave it as is.

After you set the date and time, the DOS shows its command prompt -- or it shows its command prompt and loads the application program you are using. The following sections describe the command prompt and application programs.

2.3 THE COMMAND PROMPT

The DOS command prompt is a letter followed by >. For example:

```
A>
```

is a command prompt. The command prompt tells you the DOS is ready to accept a **command** -- or an instruction -- from you.

The DOS command prompt always means the DOS is loaded and ready to use.

2.4 APPLICATION PROGRAMS

The DOS and application programs are both computer **programs.** A program is a list of instructions to the computer to do certain jobs. The DOS performs the basic functions used by all other programs. Application programs perform tasks such as word processing or inventory control. The word "application" means that the program applies the computer to a particular job. Computer programs are collectively called **software.**

You must load an application program before you can use it. When you load the program, the computer transfers it from a diskette or hard disk into the processor unit.

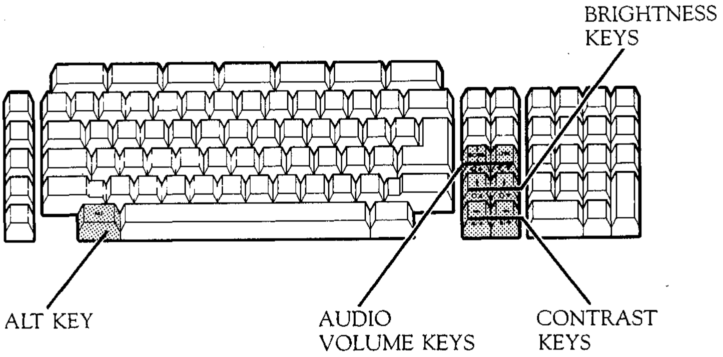
Some application programs are set up to be automatically loaded by the DOS. The DOS loads them after it displays its command prompt. You must load other application programs yourself. A program's user guide describes how to load it.

2.5 ADJUSTING BRIGHTNESS, CONTRAST, AND VOLUME

You can adjust screen brightness, contrast, and audio volume any time after the DOS is loaded. Adjust the audio volume for application programs that use sounds -- such as beeps or voice prompts -- to communicate with you.

You adjust brightness, contrast, or audio volume by holding down the ALT key and pressing one of the keys shown in Figure 2-3.

Figure 2-3: Brightness/Contrast/Volume Keys and and the ALT Key



Each brightness, contrast, or volume key has an appropriate symbol on its front key label. The label also shows an upward-pointing triangle (for "increase") or a downward-pointing triangle (for "decrease").

For example, the label on the front of the brightness-increase key has a sun symbol and a downward-pointing triangle.

You can practice adjusting the brightness:

- To increase brightness, hold down the ALT key and press the brightness-increase key one or more times.
- To decrease brightness, hold down the ALT key and press the brightness-decrease key one or more times.

Use the contrast and volume keys in the same way.

2.6 USING THE RESET BUTTON

The **reset button** is the square black button on the rear of the processor unit (Figure 2-4). When you press the reset button, the computer:

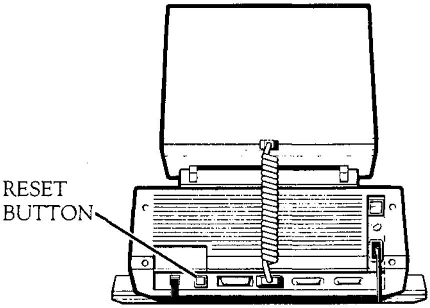
- Reloads the DOS if your system diskette is inserted.
- Displays the power-on message (Figure 1-14) if the drives are empty and the drive doors are open.

If the drives are empty but one of the doors is closed, the computer displays an error message. Open the drive door to solve this problem. (Error messages are listed and described in Chapter 22.)

Use the reset button to load a new operating system, or to recover from a problem with the computer that cannot be corrected in another way.

CAUTION: When you press the reset button, the computer may destroy the work you were doing. Do not use the reset button unless you have saved your work.

Figure 2-4: Reset Button



3. MAKING A BACKUP DISKETTE

The first work you should do with your computer is to make a DOS **backup diskette.** A backup diskette is a duplicate of another diskette.

Backup is very important. Diskettes can be damaged, and they wear out over time. If you do not keep backups, you may lose the information recorded on your diskettes.

When you use your application program, you will record your work on diskettes. Backing up these "working" diskettes is as important as backing up your system diskettes.

You use a **utility** to make a backup diskette. A utility is a program supplied with the DOS. Use the utilities DCOPY or SDCOPY to back up your system diskettes or your working diskettes. DCOPY is used with computers that have two floppy drives; SDCOPY is used with computers that have one floppy drive. DCOPY or SDCOPY (or, in some cases, both DCOPY and SDCOPY) are contained on the MS-DOS and CP/M-86 system diskettes.

It is also important to back up the information you store on the hard disk. You use the ARCHIVE utility to backup the hard disk. Section IV -- Hard Disk Utilities -- describes how to use ARCHIVE.

3.1 BACKUP WITH DCOPY

Use DCOPY only if your computer has two floppy disk drives.

You need a new diskette to make a backup diskette. Be sure to use a double-sided diskette if the diskette you are copying is double-sided.

3.1.1 BACKING UP A SYSTEM DISKETTE

To back up a system diskette with DCOPY:

1. Load the DOS (if it is not already loaded).

2. Insert the new diskette in the right drive.

3. At the DOS command prompt, type:

   dcopy

   and press Return. If you make a mistake, correct it with the Backspace key or press Return and try again.

4. The DCOPY sign-on message appears at the top of the screen:

   Diskette COPY Utility - Version n.n

   At the bottom of the screen, DCOPY asks you:

   Copy from FLOPPY drive? (A or B; press return key to end.)

5. With DCOPY, the left drive is drive A, and the right drive is drive B. Your system diskette should be in drive A. Type:

   a

6. DCOPY tells you:

   Copy from FLOPPY drive A to FLOPPY drive B.
   Press space bar when ready.

7. Press the Space bar to start the copy. (Pressing any other key cancels the copy.)

8. During the copy, the number of the track being copied appears at the bottom of the screen. The number changes each time DCOPY moves to a new track. A track is a circular section of a diskette; your diskettes have 8Ø tracks on a side.

9. DCOPY tells you when the copy is complete and asks if you want to repeat the process:

   Copy from FLOPPY drive A to FLOPPY drive B complete.

   Copy from FLOPPY drive? (A or B; press return key to end.)

1Ø. Remove the backup diskette from drive B, label it, and store it.

   CAUTION:

   - Write on the label with a soft felt tip pen. DO NOT use a ball point or pencil.

   - Store diskettes away from magnetic fields generated by telephones, magnets, or other office equipment.

11. Repeat steps 5 through 1Ø to make another backup system diskette.

12. Press Return to return to the DOS.

3.1.2 BACKING UP OTHER DISKETTES

To back up a diskette that does not contain the DOS:

1. Load the DOS with your system diskette.

2. Type:

   dcopy

   and press Return.

3. You no longer need your system diskette after the DCOPY sign-on message appears. Remove your system diskette and insert your working diskette in drive A.

4. Insert a new diskette in drive B.

5. Repeat steps 4 through 11 from the previous section.

6. To return to the DOS:

   a. Remove your working diskette from drive A.

   b. Insert your system diskette in drive A.

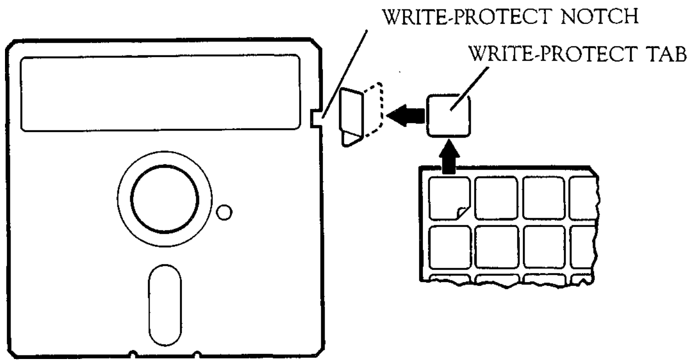
   c. Press Return.

3.2 WRITE-PROTECTING DISKETTES

After you back up the DOS, **write-protect** the backup diskette to safeguard the information on it. With **write-protection**, your computer cannot **write** -- or record -- over the information on a diskette.

Each diskette has a **write-protect notch** (Figure 3-1). To write-protect a diskette, cover the notch with an adhesive tab (Figure 3-1). Write-protect tabs are usually supplied with new diskettes.

Figure 3-1: Write-Protecting a Diskette



3.3 BACKUP WITH SDCOPY

Use SDCOPY if your computer has one floppy disk drive.

You need a new diskette to make a backup copy. **YOU MUST FORMAT THE NEW DISKETTE BEFORE YOU CAN COPY ONTO IT.** Chapter 4 describes how to format a diskette.

Be sure to use a double-sided diskette if the diskette you are copying is double-sided. You can copy a single-sided diskette onto a double-sided diskette, but this transforms the double-sided diskette into a single-sided diskette.

To back up a diskette with SDCOPY:

1. Load the DOS (if it is not already loaded).

2. At the DOS command prompt, type:

   sdcopy

   and press Return. If you make a mistake, correct it with the Backspace key or press Return and try again.

3. The SDCOPY sign-on message appears at the top of the screen. At the bottom of the screen, SDCOPY asks you to insert the source diskette (the diskette you are copying):

   Insert source diskette, then press Space bar (Return key to exit).

4. Press the Space bar if the floppy drive contains the diskette you want to copy (the source diskette).

5. If the floppy drive does not contain the diskette you want to copy:

   a. Remove the diskette in the drive.

   b. Insert the source diskette.

   c. Press the Space bar.

6. SDCOPY tells you how many times you must reinsert the source and destination diskettes to complete the copy:

   X insertions of your source and destination disks will be required.

   (where X represents a number.)

   Also, SDCOPY starts the copy process:

   Partial read #1 of X

7. SDCOPY ask you to insert the destination diskette (the diskette which will contain the copy):

   Insert formatted destination disk, then press Space bar (Return key to exit).

8. Remove the source diskette, insert a newly formatted diskette, and press the Space bar.

9. SDCOPY copies the first section of the source diskette onto the destination diskette:

   Partial write #1 of X

   and asks you to reinsert the source diskette:

   Insert source disk, then press space bar (return key to exit).

10. Continue to reinsert your source and destination diskettes until the copy is complete.

11. DCOPY notifies you when the copy is complete and prompts you to repeat the process:

    SDCOPY completed.

    Insert source disk, then press space bar (return key to end)

12. Repeat steps 1 through 11 to make another copy.

13. Press the Return key to return to the DOS.

4. PREPARING NEW DISKETTES FOR USE (FORMAT)

The work you do with the computer is recorded on floppy diskettes or on the hard disk. Before you can record information on a floppy diskette, you must prepare it to accept data.

You use the utility **FORMAT** to prepare a diskette. You can use FORMAT with computers that have one or two floppy drives.

FORMAT is contained on the MS-DOS and the CP/M-86 system diskettes and can be used to format single- or double-sided diskettes. The MS-DOS version of FORMAT prepares diskettes for use with MS-DOS or with programs that use MS-DOS. The CP/M-86 version prepares diskettes for use with CP/M-86.

CAUTION: FORMAT erases information already recorded on the diskette. Be sure the diskette you format does not contain information you need.

This chapter describes basic use of FORMAT. You can expand FORMAT, as described later in the sections on the MS-DOS and CP/M-86 operating systems.

4.1 USING FORMAT WITH TWO FLOPPY DISK DRIVES

To use FORMAT with two floppy drives:

1. Load the DOS (if it is not already loaded).

2. At the DOS command prompt:

   format

   and press the Return key. The Return key is labeled RETURN or ↵ .

- If the computer has double-sided disk drives, the FORMAT program will ask:

  FORMAT BOTH SIDES OF DISKETTE? (Y/N)

  Type **y** for "yes" or **n** for "no," and press Return.

If you make a mistake, correct it with the Backspace key or press Return and try again.

3. The FORMAT sign-on message appears at the top of the screen:

   Diskette FORMAT Utility - Version n.n

   At the bottom of the screen, FORMAT asks you:

   Format FLOPPY drive? (A or B; press return key to end.)

   DO NOT RESPOND until step 5.

4. Insert the diskette to be formatted in the right drive.

5. With FORMAT, the right drive is drive B, and the left drive is drive A. The diskette to be formatted should now be in drive B. Type:

   b

6. FORMAT tells you:

   Format FLOPPY drive B. Press space bar when ready.

7. Press the Space bar to start the format. (Pressing any other key cancels the format.)

8. During the format, the number of the track being formatted appears at the bottom of the screen. The number changes each time FORMAT moves to a new track.

   A track is a circular section of a diskette. Your diskettes have 8Ø atracks on a side.

9. FORMAT tells you when it is done and asks you to repeat the process:

   Format FLOPPY drive B complete.

   Format FLOPPY drive? (A or B; press return key to end.)

1Ø. Remove the formatted diskette from drive B and store it safely.

11. To format another diskette, repeat steps 4 through 1Ø.

12. To return to the DOS, press Return.

4.2 USING FORMAT WITH ONE FLOPPY DISK DRIVE

Use the procedure for formatting a diskette with two floppy drives EXCEPT THAT:

o At step 5, remove your system diskette from the drive and insert the diskette to be formatted. With FORMAT, the single drive is drive B.

o After you have finished using FORMAT, reinsert your system diskette and press Return to return to the DOS.

5. USING THE KEYBOARD

You can use your computer with several different keyboard models. These keyboards are shown in Appendix C.

5.1 KEY LABELS

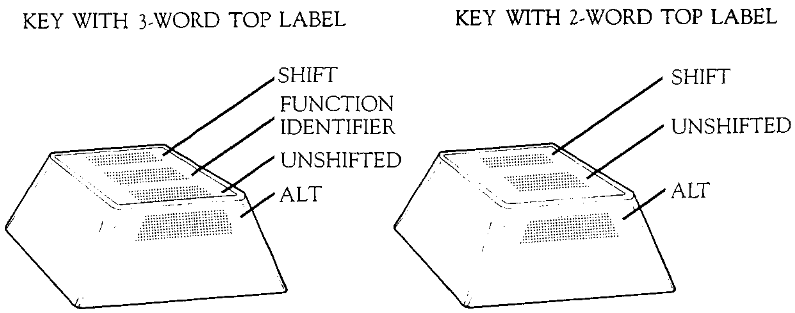
Key-top labels show shifted and unshifted key functions (Figure 5-1).

If the top label contains two words, the first word describes the shifted function, and the second word describes the unshifted function. If the top label contains three words, the second word identifies the function, the first word shows the shifted function, and the third word shows the unshifted function (Figure 5-1).

For example, the key labeled OFF/UNDL/ON is the underline key. When you press it unshifted, it turns on underline mode; everything that appears on the screen is underlined. When you press it while holding down the Shift key, it turns off underline mode. (NOTE: the underline mode is a temporary mode. It only affects what is on the screen. You cannot use it to print underlined words.)

Key-front labels show ALT (ALTernate) key functions (Figure 5-1). An ALT key function is done by pressing the key while holding down the ALT key.

Figure 5-1: Key Label Positions: Unshifted, Shift, and ALT



Some key labels on some keyboards use symbols. Table 5-1 describes the symbol labels. (Your keyboard may not contain all the labels described in Table 5-1.)

Table 5-1: Key-Label Symbols

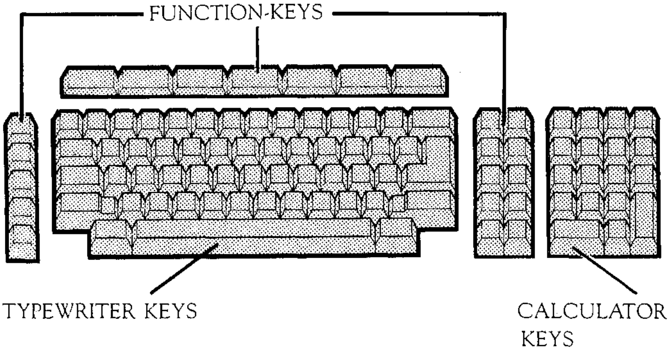
LABEL	FUNCTION
↵	Return
⇧	Shift
⇳ or 🔒	Shift lock/lock release
←	Backspace
⇥	Tab right
↑, ↓, →, ←	Cursor control
☼▲, ☼▼	Brightness control
◐▲, ◐▼	Contrast control
🔈▲, 🔈▼	Audio control

5.2 KEY FUNCTIONS

The keyboard is normally used as explained here. The use of some keys, however, varies from program to program. Refer to your program's user guide for specific key use.

The keyboard contains five main key groups (Figure 5-2): the **typewriter keys**, three groups of **function keys**, and the **calculator keys.**

Figure 5-2: Key Groups



The typewriter keys are used mainly as a typewriter. But computers -- unlike typewriters -- are particular about receiving the correct character. For example, do not type lowercase L's for numeral 1's, or uppercase letter O's for zeros.

The two typewriter keys not found on a normal typewriter are the ALT key and the Pause or Pause/Cont key:

- The ALT key is always used with another key to perform ALT key functions. To do an ALT key function, hold down the ALT key and press the other key. Most ALT-key functions are defined by the program in use.

- Use of the Pause key is defined by the program.

The top row of function keys are **general function keys.** Their use is defined by the program you are using.

The remaining function keys are **specific function keys.** They are intended to do what is shown on their labels. (Their use may also vary with the program you are using.)

5.3 THE CALCULATOR

The calculator keys are used for numeric entry or as a calculator. You can use the calculator keys to add, subtract, multiply, and divide.

You load the calculator program to use the calculator. Once the program is loaded, you can use it at any time. You can interrupt your work to perform a calculation and then return to your work exactly where you left off.

5.3.1 LOADING THE CALCULATOR PROGRAM

To load the calculator program:

1. Get to the DOS command prompt. (If you are using another program, refer to its user guide for instructions.)

2. Type:

   calc

   and press Return.

3. The DOS displays a message telling you it has installed (loaded) the calculator program.

Once you have loaded the calculator program, you turn it on to use it. You can turn on the calculator at any time.

5.3.2 TURNING THE CALCULATOR ON AND OF

Turn on the calculator by pressing a shifted Calc (press the Calc key while holding down the Shift key). The Calc key is labeled CALC or MODE/CALC.

Turn off the calculator by:

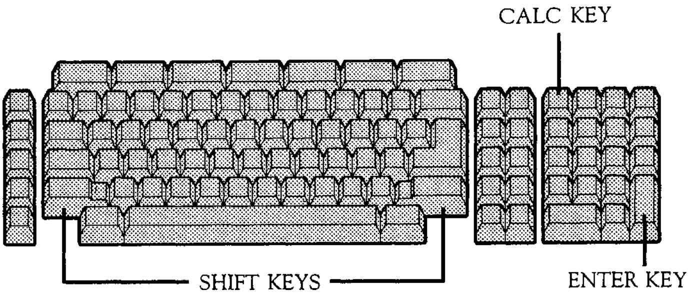
- Pressing another shifted Calc or =.

- Pressing the Enter key.

Both keys return you to your work exactly where you left off. The Enter key, however, also places the result of your calculation in your work, as if you had typed the number without having used the calculator.

Figure 5-3 shows the Calc key, the Enter key, and the Shift keys.

Figure 5-3: Calc, Enter, and Shift Keys



When you turn on the calculator, the **calculator field** appears at the bottom of the screen. The calculator field looks like this:

Calculator Functions On ! Ø. !

The exclamation points mark where the calculation appears.

Table 5-2 summarizes how to turn the calculator on and off.

Table 5-2: Turning the Calculator On and Off

FUNCTION	ACTION	RESULT
Turn calculator on	Shifted Calc key	Calculator field appears.
Turn calculator off	Shifted Calc key	Calculator field disappears.

FUNCTION	ACTION	RESULT
		o You are returned to your work where you left off.
	Enter key	o Calculator field disappears. o You are returned to your work where you left off. o The result of your calculation is placed in your work.

5.3.3 USING THE CALCULATOR

The computer shows you each step in a calculation, one number at a time. The field accepts up to 14 digits.

Press the Calc key (without the Shift key) to complete a calculation. Press the Calc key again to return the field to zero. You can start a new calculation after the first or second time you press the Calc key.

The following describes how to add, subtract, multiply, and divide with the calculator:

o Addition and subtraction:

  - Type the numbers you are adding or subtracting in any order.

  - Press the Add key (+) after each number you are adding.

  - Press the Subtract key (-) after each number you are subtracting.

  - Press the Calc key to complete the calculation.

- Multiplication:

  - Type the numbers you are multiplying, pressing the Multiply key (X) after each one.

  - Press the Calc key to complete the calculation.

- Division:

  - Type the numbers you are dividing (in order), pressing the Divide key after each one.

  - Press the Calc key to complete the calculation.

5.3.4 CORRECTING ERRORS

You can use the Backspace key and the Calc key to correct errors.

Press the Backspace key to delete the last number you typed. Press the Calc key (one or more times) to cancel a calculation and return the calculator field to zero.

6. USING YOUR COMPUTER

You are now ready to start using application programs. To use an application program, refer to the program's user guide. To use the DOS, refer to the section on MS-DOS or CP/M-86.

This chapter contains general information that applies to all programs. More detailed information can be found in the program user guides.

6.1 HARD DISK VOLUMES

The hard disk is divided into **volumes.** A volume is used much like a floppy drive with a diskette in it.

Each volume has a letter name, just like the floppy disk drives. The volumes are named when the hard disk is set up. (The program for setting up the hard disk is described in Section IV -- Hard Disk Utilities.)

With a computer that has a hard disk, the floppy drive(s) are named when the volumes are named. The names of the drive(s) depend on how the hard disk is set up. (The floppy drives on a computer without a hard disk are always named A and B.)

The volume that contains the DOS is called the primary boot volume.

6.2 THE DEFAULT DRIVE OR VOLUME

The **default drive** is the floppy disk drive or hard disk volume you are now using. The default drive is also called the **logged** drive.

The default floppy drive name or volume name is always shown in the DOS command prompt. The prompt

A> shows you that drive A is the default; B> shows you that drive B is the default.

You can move from one drive or volume to another. When you move to another floppy drive, it must contain a diskette. When you move to another hard disk volume, it must already have been set up.

To change the default while using the DOS, type the new drive or volume name, a colon (:), and press Return.

The following example shows moving from drive A to drive B and back to drive A. In the example, what you type is underlined, and (cr) represents the Return key.

```
A>b: (cr)
B>a: (cr)
A>
```

6.3 THE FILE SYSTEM

The computer stores information in **files.** These are similar to files in an office filing system. A file contains information, and each file has a name.

The user guide for an application program describes how to create and store files with that program.

6.3.1 THE FILE DIRECTORY

Each diskette or virtual volume has a **file directory.** The file directory lists the files stored on the diskette or volume. Each time you create or delete a file, the computer adds it to or deletes it from the directory.

You can list the directory for your system diskette or hard disk volume. Get to the DOS command prompt and type:

```
     dir
```

Then press Return. With MS-DOS, the DOS lists:

- Each file's name and extension (described in the next section).
- The number of bytes in each file.
- The date and time the file was created or last modified.

6.3.2 FILE SPECIFIERS

Each file is named with a **file specifier.** The computer uses the file specifier to identify the file.

A file specifier has one to three parts:

- A **file name** that describes the file. A file name has up to 8 characters.
- An optional **file extension** that usually describes the file type. An extension has up to 3 characters.
- An optional **drive designation** (a drive name and a colon). The drive designation names the drive or volume containing the file.

The drive designation locates the file for the computer. If a file specifier does not contain a drive designation, the computer assumes the file is on the default drive.

The three parts are joined as follows:

D:FILENAME.EXT

where D: is the drive designation, and .EXT is the file extension.

The sections on MD-DOS and CP/M-86 describe the specifics of their file systems in greater detail.

Each application program's user guide usually describes how it uses file specifiers.

6.4 COMMUNICATING WITH THE COMPUTER

You type a command to tell the computer to do something. Commands can be typed in upper- or lowercase. End a command by pressing Return. Each program has its own set of commands.

The computer **reads** (retrieves) information from diskette or the hard disk or **writes** (records) information while processing commands. The in-use light by a floppy drive lights each time the computer uses the drive for a read or write operation. Never open the drive door while the in-use light is on.

Most commands can be modified with **command parameters.** A command parameter changes the effect of a command. For example, the DIR command (the DOS command for displaying the file directory) can be modified with a drive designation. If your computer has two floppy drives, both with diskettes in them, you can type:

dir b:

and press Return. The computer displays the directory for drive B, rather than for drive A. If drive B is empty, the computer displays an error message to indicate a problem.

6.5 LOADING PROGRAMS

You must load a program before you can use it. When you load a program, the computer transfers the information in the program from the diskette or volume that contains the program into the processor unit. This guide describes how to load the operating systems and their utilities. Each program's user guide describes how to load it. In general, you load a program by typing its name.

You can load a program that is not on the default drive by adding a drive designation to the program name. For example, if drive A is the default drive and the DCOPY program is on drive B, you can load DCOPY by typing:

b:dcopy

and pressing Return. When you finish using DCOPY, the computer returns you to drive A.

6.6 CLEANING THE COMPUTER

Clean the computer cabinet with a soft cloth dampened with a mild household cleaner.

Clean the screen with the CRT Cleaning Cloth supplied with your computer. Do not use cleaning solution on the screen. If you do, residue from the solution will build up on the screen and obscure vision.

CAUTION:

- Never spray water or soap into the drives or onto the back of the processor.
- Never spray the screen with cleaning solution.

- Never poke anything at the screen.

- Never try to open the cabinet.

6.7 ENVIRONMENTAL AND ELECTRICAL REQUIREMENTS

Environmental requirements:

- 40 - 104 degrees Fahrenheit

- 5% - 80% relative humidity

Be sure to keep the computer out of direct sunlight.

Electrical requirements:

o	Voltage:	U.S. and Canada	-	115 Volts AC plus or minus 20% at 60Hz
o		International	-	220 Volts AC plus or minus 20% at 50Hz
o	Current:	U.S. and Canada	-	2.0 Amp
		International	-	1.0 Amp

SECTION II

THE MS-DOS OPERATING SYSTEM

7. MS-DOS SYSTEM STRUCTURE

The MS-DOS operating system provides a way for you to communicate with your computer. MS-DOS must be loaded for you to use the application programs that operate with MS-DOS. Like other operating systems, MS-DOS manages the basic functions used by these programs.

Chapters 7 through 1Ø describe how to use MS-DOS files, commands, and utilities. You can use MS-DOS commands or utilities whenever you are using the operating system. You know you are using the operating system when you see the DOS command prompt -- a drive letter name followed by >.

MS-DOS is currently the only operating system you can use with the hard disk. (Refer to Section IV -- Hard Disk Utilities, which describes the utilities for setting up and using the hard disk. The hard disk CP/M-86 emulator, which allows you to use the hard disk with programs based on CP/M-86 is described in Chapter 8.)

7.1 SECTION CONVENTIONS

Examples are shown in boldface, and what you enter is underlined. _ represents the cursor.

In examples and command-format illustrations, (cr) represents the Return key, (space) represents the Space bar, ESC represents the Escape key, and ^ represents the ALT key. Escape sequences and ALT sequences are hyphenated -- for example, Escape S is shown as ESC-S and ALT C is shown as ALT-C.

In command formats, command names are in upper case, and command parameters are in lower case. (You may enter commands and command parameters in upper or lower case at the keyboard.)

In command formats, brackets ([]) indicate optional item(s). A vertical bar (|) separates alternatives. An ellipsis in brackets ([...]) indicates that the preceding parameter may be repeated. General forms appear as single words or single-word abbreviations -- such as filename, drivename, filespec, and ext.

The terms system diskette and DOS diskette refer to any diskette that contains the MS-DOS operating system.

7.2 SYSTEM RESOURCES

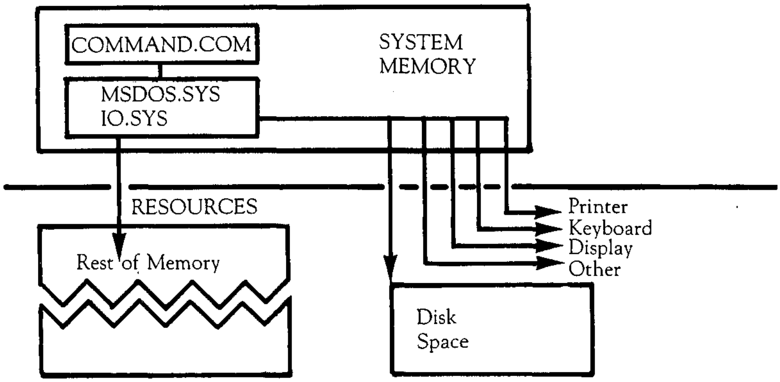
MS-DOS, itself, consists of three files:

```
COMMAND.COM
MSDOS.SYS
IO.SYS
```

These three files control all system resources. Note that MSDOS.SYS and IO.SYS are "hidden" files that are not displayed in the diskette file directory.

The relationship between these three files and system resources is shown in Figure 7-1.

Figure 7-1: MS-DOS and Its Resources



The system's most important resources -- its disk space and its memory — are described in the following sections.

7.2.1 DISK SPACE

In MS-DOS, disk space is divided into four parts:

o The reserved sectors contain information used each time MS-DOS is loaded.

o The directory contains information about each file on a given diskette. This information includes the file's complete file specifier (file specifiers are described in Section 7.3), its size, and the time and date of its last modification.

o The file allocation table (FAT) contains location information for the data making up each file on a diskette.

o Files occupy the great majority of disk space. An individual file does not necessarily reside in contiguous sectors on the diskette; its contents may be "scattered" on the diskette to decrease wasted disk space.

7.2.2 MEMORY

MS-DOS controls main memory as well as disk space and other devices. MS-DOS can load files into memory as data files or as executable files. The actual loading of files is performed by MSDOS.SYS, the lowest level of the MS-DOS operating system. COMMAND.COM supervises the loading of executable files.

Most well-designed programs return control to MS-DOS after normal or abnormal termination of the program.

Part of COMMAND.COM may be overlaid to make room for a particularly large executable file. After such a file is executed, MS-DOS automatically loads the overlaid part of COMMAND.COM back into system memory, and normal execution of COMMAND.COM resumes. COMMAND will attempt to reload the transient portion from the default drive at startup, not from the current default drive. If COMMAND does not exist on the startup default drive, the recovery process allows specification of a new COMMAND.COM drive. This drive will be used for this and subsequent reloads of the transient portion.

If MS-DOS finds an incorrect version of COMMAND.COM when it attempts to load the overlaid section, it displays a similar message:

```
Invalid COMMAND.COM
Insert DOS disk in default drive and strike any key
when ready
```

7.3 THE FILE SYSTEM

MS-DOS stores information on diskettes in the form of files; a file is one or a group of related characters. A single file can be any length, up to the data storage capacity of a diskette (6ØØ Kbytes per side), or of a hard disk volume.

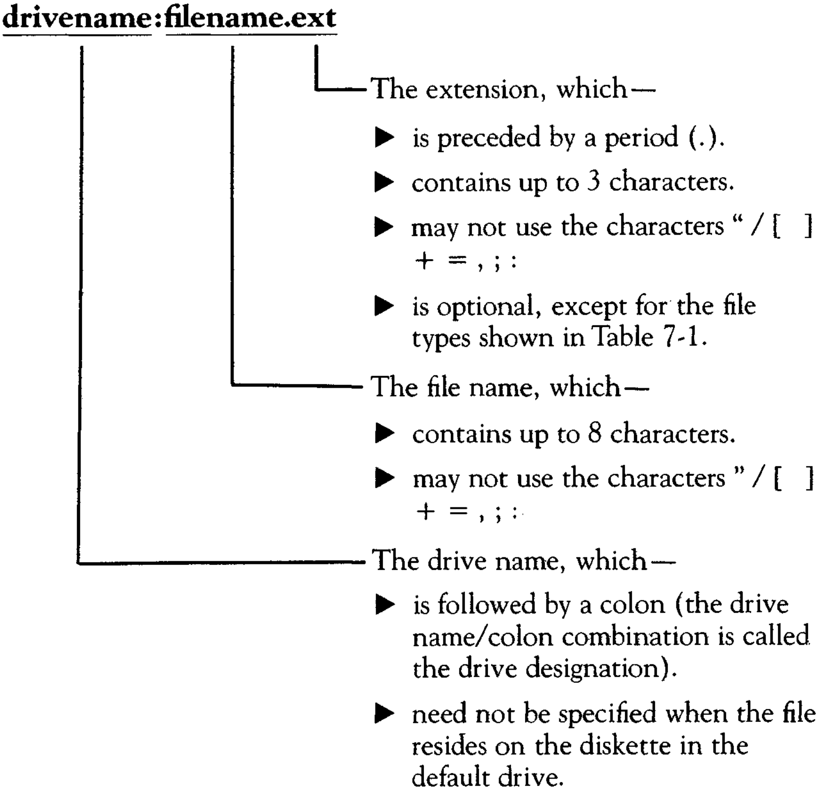
The file system may be thought of as the external organization of system resources. MS-DOS supports "device independent I/O," which means that the distinction between files and devices is internal only. Therefore, you can treat devices as files and refer to either with "file specifiers" (as described in 7.3.1).

7.3.1 FILE SPECIFIERS

A file is identified by its file specifier. A file specifier consists of one to three parts: a file name and (optionally) a file extension and/or a drive designation. If the drive designation is omitted from a file specifier, MS-DOS assumes that the file is on the diskette in the default drive.

The three parts of a file specifier are illustrated in Figure 7-2.

Figure 7-2: File Specifier Structure



drivename:filename.ext

The extension, which—

- is preceded by a period (.).
- contains up to 3 characters.
- may not use the characters " / [ ] + = , ; :
- is optional, except for the file types shown in Table 7-1.

The file name, which—

- contains up to 8 characters.
- may not use the characters " / [ ] + = , ; :

The drive name, which—

- is followed by a colon (the drive name/colon combination is called the drive designation).
- need not be specified when the file resides on the diskette in the default drive.

7.3.2 MS-DOS FILE EXTENSION CONVENTIONS

MS-DOS automatically defines the meanings of several file extensions. The file extensions defined by MS-DOS are presented in Table 7-1.

Extensions other than those automatically defined by MS-DOS or your application program can be assigned at your discretion. For example, you might use the extension .TXT for all text files.

Table 7-1: MS-DOS File Extension Conventions

FILE EXTENSION	MS-DOS INTERPRETATION
.ASM	8Ø86 assembly language source code
.BAK	Backup file created by application program
.BAS	BASIC source code (Microsoft BASIC)
.BAT	Batch command file
.COB	COBOL source code
.COM	Executable command file
.DAT	Data file (assumed to be ASCII)
.EXE	Relocatable executable file
.FOR	FORTRAN source code
.INT	Intermediate compiledcode
.LIB	Library file
.LST	Listing of compilation or assembly
.MAP	List file from linker
.OBJ	Relocatable object code module
.OVR	Overlay module
.PAS	Pascal source file
.PRN	Listing of compilation or assembly
.CRF	Cross-reference
.$$$	Temporary system-generated file

7.3.3 WILD-CARD CHARACTERS

MS-DOS wild-card characters allow you to use a single command to affect a group of files. Wild-card characters are similar to wild cards in card games in that the wild-card characters take on the meaning that you assign to them.

The two MS-DOS wild-card characters are the question mark (?) and the asterisk (*).

The ? wild-card character means "match any character -- or no character -- in this particular location in the file name or file extension." For example, PAY-???.ROL matches each of these file specifiers:

```
PAY-JAN.ROL
PAY-FEB.ROL
PAY-MAR.ROL
PAY-JL.ROL
```

The * wild-card character means "pad with ?s." This enables you to refer to entire families of files. For example, *.BAS refers to all files with the extension .BAS. WS*.* refers to any file whose file name (no matter how long) starts with WS, regardless of extension.

The * character pads all possible characters that follow it, within the file name or the extension where it appears. This means that MS-DOS cannot read or match any characters following the * in a file name or extension. For example, *86.EXE matches all files with extension .EXE (????????.EXE), not just files whose names end in 86 and have the extension .EXE.
Wild-card characters can be used in combination. For example, ?TEST.* refers to all files whose file name is four or five characters long, the last four of which are TEST, regardless of file extension.

Note that the file specifier *.* refers to all files.

7.3.4 DEVICE FILE NAMES

MS-DOS uses certain 3-letter file names for the names of devices:

- AUX refers to input from or output to an auxiliary device.
- CON refers to keyboard input or output to the terminal screen.
- LST refers to the printer.
- NUL does not refer to a particular file or device. NUL is used when the syntax of a command requires an input or output file name.

Even when these file names are given drive designations or file extensions, they remain associated with the devices listed above. Thus, CON refers to the terminal console, not a disk file. This results from MS-DOS's device independent I/O, and permits you to treat devices as if they were files.

7.3.5 THE FILE DIRECTORY

A diskette's file directory contains information about each file on the diskette. This information includes the file's complete file specifier, its size, and the time and date of its last modification. When you create or delete a file, MS-DOS automatically adds its name to or deletes its name from the directory.
The DIR command is used to display a diskette's file directory (as described in Chapter 9).

7.4 COMMAND LINE EDITING

The command line is created and altered by entering --

Alphanumerics
Punctuation
Special editing functions
Alternate-character functions

You can use the special editing and alternate-character functions whenever terminal input is required. These functions are always resident in the operating system.

The model for command line input is as follows:

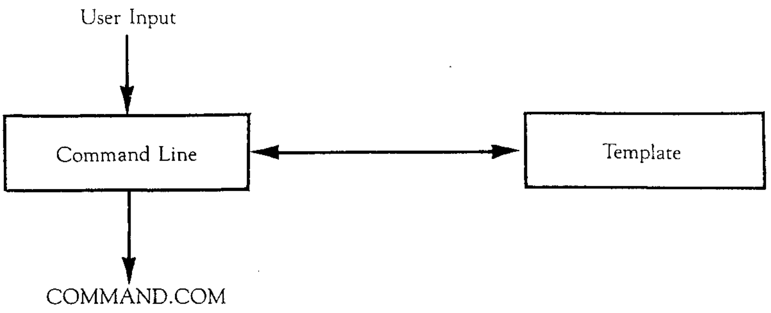
- When text is entered from the keyboard, it is held in the command line buffer until the Return key is pressed.

- A Return sends the contents of the command line to COMMAND.COM for processing.

- A Return also copies the command line to the "template," another buffer intimately related to the command line buffer.

The relationship between the command line and the template is shown in Figure 7-3.

Figure 7-3: The Command Line and the Template



With the MS-DOS special editing functions, you can use the template to --

- o Instantly repeat a command line in two key strokes.
- o Edit and retry an erroneous command line, without reentering the entire command line.
- o Edit and execute a command line similar to a preceding command line with a minimum of typing.

7.5 SPECIAL EDITING FUNCTIONS

Table 7-2 lists the MS-DOS special editing functions. These functions are accomplished by entering the escape sequences listed in the table (and the additional specified character, or characters, if necessary). The function keys on your keyboard can also be programmed to perform the special editing functions. Your dealer can provide you with a **Programmer's Tool Kit** that allows such use of the function keys.

Table 7-2: Special Editing Functions

Dos 2.11	FUNCTION NAME	ESCAPE SEQUENCE*	EDITING EFFECT
F3	C1	Escape S	Copies one character from the template to the command line.
F6	CM "C"	Escape T	Copies all characters up to the specified character "C" from the template to the command line (where "C" is any character).
F7	CT	Escape U	Copies all remaining characters in the template to the command line.
F4	S1	Escape V	Skips over (does not copy) a character in the template.
F5	SM "C"	Escape W	Skips over (does not copy) the characters in the template up to the specified character "C", where "C" is any character.
	QI	Escape H	Voids the current input; leaves the template unchanged.
F1	INS	Escape P	Enters insert mode.
F1	REP	Escape Q	Exits insert mode; the REP mode is the default mode.
F2	NT	Escape R	Makes the new line the new template.

* The escape sequences listed in this column are accomplished by pressing the Escape key and then pressing the specified letter key. The letter keys must be typed in upper case.

7.6 ALTERNATE-CHARACTER FUNCTIONS

Seven alternate characters may be used while entering command lines. These alternate characters and their functions are shown in Table 7-3.

Table 7-3: Command-Line Alternate Character Functions

ALT SEQUENCE	FUNCTION
ALT-C	Aborts current command.
ALT-H or Backspace	Removes last character from command line, Backspace and erases character from terminal screen.
ALT-J	Inserts physical end-of-line, but does not empty command line. Uses line feed to extend the current logical line beyond the physical limits of one terminal line.
ALT-N	Cancels echoing of output to the printer.
ALT-P	Echoes terminal output to the printer.
ALT-S	Suspends display of output to terminal screen. Console (and printer, if output is being echoed to the printer) remains frozen until any key is pressed.

ALT-X Cancels the current line, empties the command line, and outputs a backslash (\), carriage return, and line feed. The template is not affected.

7.7 COMMAND INTERPRETATION

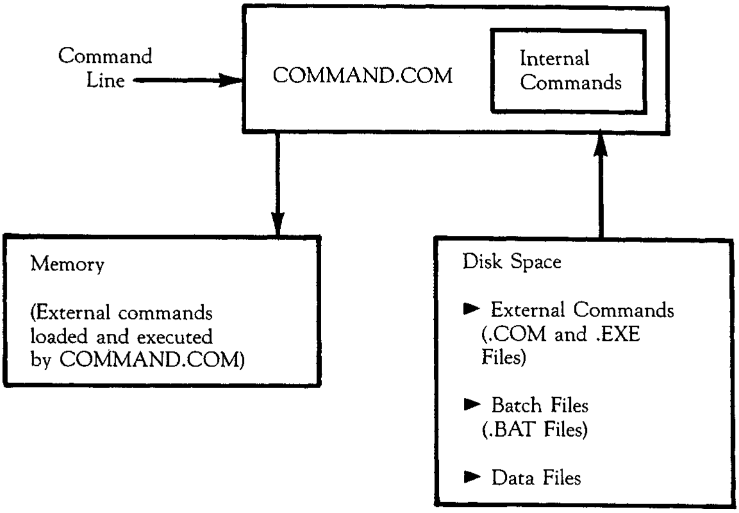
COMMAND.COM processes and interprets MS-DOS commands. Command interpretation begins when COMMAND.COM scans the command line for the name of a legal command.

Commands themselves are of two types: internal and external. Internal commands are part of COMMAND.COM; they are loaded into memory when the operating system is loaded. External commands reside in files that have a .COM or an .EXE extension. They are loaded from diskette only when needed.

Batch command files (.BAT files) may contain both internal and external commands (see "Batch Commands"). As with external commands, batch files are loaded from diskette when needed.

The command interface is illustrated in Figure 7-4.

Figure 7-4: MS-DOS Command Structure


Command Line
COMMAND.COM
Internal Commands
Memory
(External commands loaded and executed by COMMAND.COM)
Disk Space
► External Commands (.COM and .EXE Files)
► Batch Files (.BAT Files)
► Data Files

COMMAND.COM provides the MS-DOS command prompt -- a one-letter drive name and a greater-than symbol (i.e., A> or B>).

7.7.1 INTERNAL COMMANDS

Internal commands are always available when COMMAND.COM resides in memory. The internal commands need not be available on disk when they are executed.

The internal commands are listed here. Their use is described in Chapter 9.

```
COPY      REM
DATE      REN
DEL       TIME
DIR       TYPE
PAUSE     TOD
          ?
```

7.7.2 EXTERNAL COMMANDS

Any file with the extension .COM or .EXE is an external command. The following external commands are described in Chapter 8:

CHKDSK.COM
COM
SYSCOPY.EXE
FORMAT.COM
RDCPM.COM
SETIO.COM
CPM.EXE
SDCOPY.EXE

7.7.3 BATCH COMMANDS

The MS-DOS batch facility allows processing of files containing "batches" of commands. The commands in batch files are processed as if they were typed at a terminal. Each batch file must be named with the execution by entering its file name (without the Return:

```
[drivename:]filename [parameter...] (cr)
```

For example, a batch file named NEWDISK.BAT might contain these commands:

```
REM THIS IS FILE NEWDISK.BAT (cr)
REM (THE .BAT EXTENSION MUST BE GIVEN) (cr)
PAUSE INSERT DISK IN DRIVE B (cr)
FORMAT B: (cr)
DIR B: (cr)
CHKDSK B: (cr)
```

To execute this .BAT file, you would simply enter NEWDISK and press the Return key. The result would be as you had entered each of the lines in the .BAT file individually.

By creating a .BAT file with prototype commands containing positional parameters, parameters may be passed to the .BAT file when it is executed. You may specify up to 1Ø positional parameters, named %Ø through %9. (Positional parameters are particularly useful when used with the MS-DOS linker and compilers).

The parameters are substituted in their order on the invocation line for corresponding occurrences in the batch file. If the dummy parameter %Ø is used, the batch facility substitutes the name of the batch command itself for parameter %Ø. Thus, you can create batch commands that can be used on more than one set of files, and that can be used to reexecute themselves.

For example, you might create a batch file for use with BASCOM, the MS-BASIC compiler, containing --

```
REM THIS IS FILE COMPILE.BAT (cr)
REM START BATCH COMPILE (cr)
BASCOM %1; (cr)
LINK %1; (cr)
```

To invoke this file, substituting the program DEMO for the positional parameters, you would enter the command --

```
COMPILE DEMO (cr)
```

The result would be as if you had entered the following --

```
REM THIS IS COMPILE.BAT (cr)
REM START BATCH COMPILE (cr)
BASCOM DEMO; (cr)
LINK DEMO; (cr)
```

When you load MS-DOS, COMMAND.COM searches for the file AUTOEXEC.BAT. If a file with that name exists, it is automatically processed, and the startup TIME and DATE commands are bypassed. If COMMAND.COM does not find AUTOEXEC.BAT, it displays the normal MS-DOS prompt.

Two MS-DOS commands are available expressly for use in batch files: REM and PAUSE. REM permits the inclusion of remarks in batch files. PAUSE prompts you with an optional message and permits you to abort execution of a batch file at a given point. REM and PAUSE are further described in Chapter 9.

8. MS-DOS UTILITIES

MS-DOS includes seven utilities for managing computer resources:

- FORMAT formats diskettes for system use and, optionally, creates a new system diskette.
- DCOPY copies a diskette's contents onto another diskette. DCOPY is used with computers that have two floppy disk drives.
- SDCOPY also copies a diskette's contents onto another diskette. SDCOPY is used with computers that have one floppy disk drive.
- SYSCOPY copies the operating system onto a diskette or hard disk volume.
- CHKDSK scans the directory of a diskette and checks for consistency.
- RDCPM transfers files form a CP/M-86 diskette onto an MS-DOS diskette.
- SETIO displays or changes the I/O byte. The I/O byte associates a logical device with a physical device.

8.1 THE FORMAT UTILITY

8.1.1 COMMAND FORMAT AND FUNCTION

The command format for the FORMAT utility is --

FORMAT [drivename:] [/C][/D][/E][/I][/Z] (cr)

where drivename is the name of the drive containing a diskette to be formatted, and /C, /D, /E, /I, and /Z are command "switches."

The FORMAT program prepares diskettes to receive data. In the process, FORMAT automatically erases any previous files on the diskette. New diskettes must be formatted before they can be used.

8.1.2 COMMAND OPERATION

When you load FORMAT (by entering the FORMAT command), the utility displays its one-line sign-on banner. While FORMAT is running, it displays the number of the track it is formatting at the bottom of the screen. (Your computer's diskettes contain 8Ø tracks on a side.)

Drive Name Parameters

If you do not designate a drive name with the FORMAT command, FORMAT prompts you --

o To enter the name of the drive containing the diskette to be formatted.

o To press the Space bar to start the format process.

o To repeat the process once the format is complete. (To exit FORMAT, you press the Return key.)

You can thus continue to format diskettes without reloading FORMAT. (After you load FORMAT, you can remove the system diskette from drive A and then use both drives for formatting diskettes.)

If you designate a drive name with the FORMAT command, FORMAT simply formats the designated diskette and exits to the operating system.

Command Switch Parameters

You can expand the FORMAT command by using five command switches:

- o /C displays the total number of soft errors found and the amount of space available on the diskette.
- o /D formats a double-sided diskette.
- o /E displays the locations of soft errors encountered, the total number of soft errors found, and the amount of space available on the diskette. The /E switch implies the /C switch.
- o /I clears the diskette's directory and file allocation tables only. You must use a diskette that has already been formatted with this switch. The switch causes the program to skip its formatter portion.
- o /Z displays disk zone information (size of tracks and gaps).

You may use FORMAT switches in combination. For example, the command --

FORMAT B: /E/D (cr)

would format both sides of the diskette in drive B and display the locations and total numbers of the soft errors encountered. (Note that if the /I switch is specified, the /D, /E, and /Z switches have no meaning and are ignored.)

Diskette Tracks, Zones, and Soft Errors

Switches /E, /C, and /Z display information about diskette tracks, zones, and soft errors. Tracks are circular sections of a diskette; your computer's diskettes have 80 tracks on a side. These tracks are grouped into eight zones; the drive motor runs at a different speed on each zone. Soft errors are worn or flawed spots on the diskette that may make it unreliable for use. The /C switch (for counting soft errors) allows you to gauge the reliability of a diskette for recording data. If 15 or more soft errors appear, repeat the FORMAT process a few times. If this number of errors persists, discard the diskette and try another.

8.1.3 COMMAND EXAMPLES

The examples in this section show FORMAT prompts and messages consecutively, as they appear on the screen. The examples do not indicate where on the screen the prompts and messages are displayed.

The following example illustrates using FORMAT without parameters.

```
A>format (cr)

Diskette FORMAT Utility -- Version msl.1-07
82.10.30

Format drive? (A or B; press return key to
end)b

Format drive B.  Press space bar when ready.
(space)

Format drive B complete.

Format drive? (A or B; press return key to end)
(cr)

A>_
```

8.2 THE DCOPY UTILITY

8.2.1 COMMAND FORMAT AND FUNCTION

The command format for the DCOPY utility is --

DCOPY [drivename: to drivename:] [/C] [/E] [/Z] (cr)

where the first drive name is the name of the drive containing the diskette to be copied from, the second drive name is the name of the drive containing the diskette to be copied to, and /C, /E, and /Z are command switches.

DCOPY copies the contents of one diskette onto another diskette, creating a literal twin of the original diskette. In the process, DCOPY formats the new diskette (eliminating the need to run FORMAT separately). NOTE: DCOPY does not work with Hard Disk Volumes.

8.2.2 COMMAND OPERATION

When you load DCOPY (by entering the DCOPY command), the utility displays its one-line sign-on banner. While DCOPY is running, it displays the number of the track it is copying at the bottom of the screen. (Your diskettes contain 8Ø tracks on a side.)

Drive Name Parameters

If you do not designate the copy-from and copy-to drive names with the DCOPY command, DCOPY prompts you --

- o To enter the name of the drive containing the diskette to be copied.

- To press the Space bar to start the copy process.

- To repeat the process once the copy is complete. (To exit DCOPY, you press the Return key).

You can thus continue to copy diskettes without reloading DCOPY. (After you load DCOPY, you can remove the system diskette from drive A and then use both drives for copying diskettes.)

If you designate the copy-from and copy-to drive names with the DCOPY command, DCOPY simply copies the designated diskette and exits to the operating system.

Unless you are copying your system diskette, use DCOPY without drive name parameters. Otherwise, the copy process will begin before you can safely remove your system diskette and replace it with the copy-to or copy-from diskette.

Command Switch Parameters

You can expand the DCOPY command by using three command switches:

- /E displays the locations of soft errors encountered and the total number of soft errors found. (Soft errors are described in 8.1.) The /E switch implies the /C switch.

- /C displays the total number of soft errors found.

- /Z displays disk zone information (size of track and gaps; tracks and gaps are described in 8.1).

You may use DCOPY switches in combination. For example, the command --

```
DCOPY B: TO A: /C/Z (cr)
```

Would copy the contents of the diskette in drive B onto the diskette in drive A and display disk zone information and the total number of soft errors found.

8.2.3 COMMAND EXAMPLES

The examples in this section show DCOPY prompts and messages consecutively, as they appear on the screen. The examples do not indicate where on the screen the prompts and messages are displayed.

The following example illustrates using DCOPY without parameters.

```
A>dcopy (cr)

Diskette COPY Utility -- Version msl.Ø-Ø5
82.1Ø.Ø7

Copy from FLOPPY drive? (A or B; press return
key to end)a

Copy from FLOPPY drive A to drive FLOPPY B.
Press space bar when ready. (space)

Copy from FLOPPY drive A to drive FLOPPY B
complete.

Copy from FLOPPY drive? (A or B; press return
key to end) (cr)

A>_
```

The following example illustrates using DCOPY with the /C and drive name parameters:

```
A>dcopy a: to b: /c (cr)

Diskette COPY Utility -- Version ms1.Ø-Ø2
82.Ø5.25

Dcopy complete.  8Ø tracks copied: Ø soft
errors.

A>_
```

8.3 THE RDCPM UTILITY

The RDCPM utility allows you to transfer files from a CP/M-86 diskette onto an MS-DOS diskette or to display the directory of a CP/M diskette while using MS-DOS.

8.3.1 TRANSFERRING CP/M-86 FILES

The command format for transferring files from a CP/M-86 diskette onto an MS-DOS diskette is --

```
[drivename:]RDCPM drivename:filename.ext (cr)
```

where drivename represents the name of the drive containing the MS-DOS diskette and drivename:filename.ext represents the file specifier of the CP/M-86 file you wish to transfer.

You may use wild-card characters in the file name and extension of the file to be transferred.

When MS-DOS has transferred all files, it displays the message --

```
File transfer complete
```

and returns control to the operating system command level.

The following example illustrates transferring all files with the file name ALPHA on the CP/M-86 diskette in drive B onto the MS-DOS diskette in drive A.

```
A>rdcpm b:alpha.* (cr)

Read CP/M Disk Utility — Version 1.7

File transfer complete

A>_
```

8.3.2 DISPLAYING A CP/M-86 DISKETTE DIRECTORY

The command format for displaying the directory of a CP/M-86 diskette while using MS-DOS is --

```
[drivename]RDCPM DIR drivename:[filename[.ext]]
(cr)
```

where drivename specifies the drive containing the MS-DOS diskette and drivename:[filename.[ext]] specifies the drive containing the CP/M-86 diskette and (optionally) the files to be listed. For example, the command —

```
A:RDCPM DIR B: (cr)
```

would display the directory of the CP/M-86 diskette in drive B. The command --

```
A:RDCPM DIR B:ALPHA.* (cr)
```

would display the directory entries for all files with the file name ALPHA on the CP/M-86 diskette in drive B.

Displaying the directory of the CP/M-86 diskette allows you to determine which files need to be transferred to the MS-DOS diskette.

8.4 THE CP/M-86 EMULATOR

The CP/M-86 emulator runs CP/M-86 programs under MS-DOS. The emulator supports the commonly used CP/M-86 functions and provides a CP/M-86 environment for the program. The program is executed as under CP/M-86; recompilation, reassembly, or other translation is not necessary.

The following sections describe how to use the emulator to run CP/M-86 programs on the hard disk (under MS-DOS) as well as the emulator program characteristics.

8.4.1 EMULATOR OPERATION

To run a CP/M-86 program on the hard disk --

1. Transfer the CP/M-86 files onto an MS-DOS diskette, using RDCPM (see 8.3).

2. Type --

   cpm

   followed by a space, the CP/M-86 command line (exactly as you would type it under CP/M-86), and a Return.

8.4.2 EMULATOR DESCRIPTION

The emulator is partitioned into modules, each of which handles a set of the CP/M-86 BDOS functions. These are --

o Emulator initialization

- o Program loader
- o Memory management
- o Run time interface
- o File system interface
- o Byte-oriented device interface
- o System functions

Emulator Initialization

The emulator is loaded and resides at high memory. Thus the TPA, as seen by the emulated program, corresponds to the CP/M-86 environment.

The input command line is formatted according to CP/M-86 CCP conventions. The specified program is loaded, the base page is initialized, and control is turned over to the program.

Program Loader

The CP/M-86 program loader is fully emulated. All three memory models are supported.

Memory Management

Memory management functions are fully emulated. However, functions implied by CCP artifacts, such as "system reset" and "stay resident", are not supported.

The memory functions emulated are:

- Get maximum available memory
- Get maximum memory, absolute
- Get memory region
- Get absolute memory region
- Free memory
- Free all memory

Run Time Interface

The run-time interface module handles the CP/M-86 BDOS interrupt and dispatches the appropriate routine. The emulator provides a stack, internal to the emulator, distinct from the program's init stack.

File System Interface

The standard CP/M-86 FCB is used for all emulations, and the CP/M-86 requests are mapped into corresponding MS-DOS requests. All user accessible FCB data is maintained by the emulator identically to CP/M-86.

The system FCB data space is reserved, but the information in it is not valid. Because of the lack of sparse files under MS-DOS, programs that use certain random write techniques may require more disk space.
Directory maintenance is functionally compatible. Maintenance functions provide a directory block with the index and all fields except the name and extent cleared to zero. The remainder of the directory block contains inactive entries.

Two CP/M-86 functions cannot be supported because they use CP/M-86 directory infomation. These functions are "set file attribute" and search with "?" drive.

The following file system interface functions are supported:

- Open/Make
- Close
- Read sequential
- Write sequential
- Read random
- Write random
- Set random record
- Write random with zero fill
- Search first/next (with exception described above)
- Delete
- Rename
- File size

Byte-Oriented Device Interface

Byte oriented logical devices are fully supported, including redirection via the I/O byte.

The following functions are supported:

- Console input
- Console output
- Reader input
- Punch output
- List output
- Direct console I/O
- Print string
- Read console buffer
- Get console status

System Functions

The system functions supported are as follows:

- System reset (terminates emulation; does not allow program to remain resident).
- Return version number.
- Reset disk system (MS-DOS disk reset).
- Select disk (uses MS-DOS function).
- Return login vector (reflects all drives known to MS-DOS).
- Return current disk (CP/M-86 equivalent).
- Get DMA address (CP/M-86 equivalent).

- Write-protect disk (marks disk as write protected within the emulator).

- Get address (R/O vector; returns address of the emulator R/O vector).

- Reset drive (resets emulator R/O vector).

- Get/Set DMA base (CP/M-86 equivalent).

The following direct BIOS calls are supported:

- INIT (terminates emulation)

- WBOOT (terminates emulation)

- CONST, CONIN, CONOUT

- LIST, LISTST, PUNCH, READER

- HOME, SELDSK

- GET/SET IO BYTE

These system functions are NOT supported:

- Get address (allocate)

- Get/Set user code

The direct BIOS calls NOT supported are as follows:

- SETTRK, SETSEC, SETDMA

- READ, WRITE, SECTRAN

- SETDMA, SETDMAB, GETSEGB

If you attempt to use these functions, an error message is displayed, and the program terminates.

8.5 THE CHKDSK UTILITY

The command format for CHKDSK is --

CHKDSK [drivename:] (cr)

where drivename represents the name of the drive containing the diskette you wish to check. If you do not designate the drive, CHKDSK checks the diskette in the default drive.

CHKDSK scans the designated diskette's directory and checks it for consistency. If CHKDSK finds an error, it tries to correct it and continues checking the diskette. After checking the diskette, CHKDSK displays the appropriate error messages (if any) and a status report on the amount and availability of diskette and memory space. (CHKDSK error messages are listed in Chapter 22.)

The following example illustrates using CHKDSK to check a diskette in drive B while using drive A:

```
A>chkdsk b: (cr)

620544 bytes total disk space
43008  bytes in 2 hidden files
278528 bytes in 16 user files
299008 bytes remain available on disk

99184  bytes total memory
87792  bytes free

A>_
```

8.6 THE SETIO UTILITY

SETIO is a utility program to display or change the I/O byte. The I/O byte associates a logical device with a physical device.

SETIO has three modes. If invoked without parameters, the assignment table is displayed. If invoked with an invalid device assignment, the command format is displayed with the assignment table. If invoked with a valid device assignment, an updated table is displayed with the new assignment.

Examples: (Highlighted fields are enclosed in brackets [].)

```
A>setio (cr) (without parameters)

Logical Device              Physical Devices

     CON                    TTY CRT BAT UL1
     AUXIN                  TTY PTR UR1 UR2
     AUXOUT                 TTY PTP UP1 UP2
     LST                    TTY CRT LPT UL1


A>setio ?(cr)   (NOTE: ? is an invalid parameter.)

SET I/O  VERSION n.n
usage:  SETIO [<logical device> = <physical
device>]

     CON                    TTY[CRT]BAT UL1
     AUXIN                 [TTY]PTR UR1 UR2
     AUXOUT                [TTY]PTP UP1 UP2
     LST                    TTY CRT LPT[UL1]
```

```
A>setio lst = tty(cr)  (valid parameters)

SET I/O  VERSION n.n

     CON                    TTY[CRT]BAT UL1
     AUXIN                 [TTY]PTR UR1 UR2
     AUXOUT                [TTY]PTP UP1 UP2
     LST                   [TTY]CRT LPT UL1

A>
```

Table 8-1: Logical and Physical Devices

DEVICE TYPE/NAME	DESCRIPTION
Logical Devices	
CON:	Console device -- the principal interactive console which communicates with the operator. Typically, CON: is a device such as a CRT or teletype.
LST:	List device -- the principal listing device; usually a hard-copy device, such as a printer or teletype.
AUX:	Auxiliary input device.
AUX:	Auxiliary output device.
Physical Devices	
TTY:	Serial output -- port A (teletype-style printer -- RS-232-C)
CRT:	Keyboard and cathode ray tube display

DEVICE TYPE/NAME	DESCRIPTION
LPT:	Parallel port printer (Centronics)
UL1:	Serial printer -- port B (RS-232-C)
BAT:	Batch mode-reader (AUX:) as input; a printer (LST:) as output.
UC1:	External console (to be developed)
PTR:	High speed read (to be developed)
UR1:	(To be developed)
UR2:	(To be developed)
PTP:	High speed punch (to be developed)
UP1:	(To be developed)
UP2:	(To be developed)

8.7 THE SYSCOPY Utility

The SYSCOPY utility is used to copy an operating system onto a diskette or virtual volume. You can then use the diskette or volume to load (boot) the operating system.

The operating system is copied from either an existing system diskette/volume or a file created by the system configuration process. (The system configuration process is described in the **Programmer's Tool Kit, Volume I**).

8.7.1 COMMAND SYNTAX

The SYSCOPY syntax is --

SYSCOPY <source> [<destination>]

where source is the diskette, volume, or file that contains the operating system, and destination is the diskette or volume onto which the operating system is to be copied.

8.7.2 COMMAND OPERATION

You can specify the source in two forms: a drive specification (e.g., A:) indicates the source is an existing system diskette or volume; a file name indicates the source is the specified operating system file (created during system configuration). If you give a file name without an extension or drive letter, the system assumes the extension SYS and the default drive.

In addition to the operating system, the source drive must contain the file COMMAND.COM. This file is copied with the operating system.

Specify the destination by giving the destination drive letter and a colon only. If you specify the destination with the command, the program terminates after the copy is complete. (Because no user interaction is necessary, you can use this form in a batch command.)

If you do not specify the destination with the command, the utility prompts you for it. After you respond to the prompt, the utility displays your response for verification. After the copy is complete, you are prompted for another destination.

8.7.3 COMMAND EXAMPLES

The following example illustrates copying the operating system and COMMAND.COM from drive A to drive B. The source is the system diskette or system volume assigned to drive A.

```
A>SYSCOPY A: B:

SYSCOPY -- Version x.x  yy.mm.dd

A>
```

The following example illustrates copying the operating-system file MSDOS.SYS and COMMAND.COM from the diskette or hard disk volume assigned to default drive (drive C) to the diskette or hard disk volume assigned to drive B.

```
C>SYSCOPY MSDOS B:

SYSCOPY -- Version x.x  yy.mm.dd

C>
```

The following example illustrates copying the operating-system file MSDOS.NEW and COMMAND.COM from the diskette or hard disk volume assigned to drive C to the diskette assigned to drive F. It also copies them to the hard disk volume (named MYVOLUME) assigned to drive B.

```
A>SYSCOPY C:MSDOS.NEW

SYSCOPY -- Version x.x  yy.mm.dd

Destination drive ? ( A-F; press return to
end.) F
Copy to F: (FLOPPY). Press space bar when
ready. <space>
Copy completed.

Destination drive ? ( A-F; press return to
end.) B
Copy to B: (MYVOLUME). Press space bar when
ready. <space>
Copy completed.

Destination drive ? ( A-F; press return to
end.) <return>
A>
```

8.7.4 ERROR MESSAGES AND RECOVERY

Bad parameter on command line -- <bad param>

The indicated parameter <bad param> is not correct. Correct the command and redo it. The most common parameter errors are --

- o No parameter specified. <source> must be on the command.

- o Invalid drive. The drive specified must be assigned.

- o Bad destination. The destination specified is not a drive.

- o Too many parameters. More than source and destination typed.

Cannot Open MSDOS.SYS

The specified source drive is not a bootable diskette or volume. Specify either a bootable volume, or a source file name.

Cannot Open <source file name>

The specified source file name cannot be opened. Make sure you typed the name correctly and that it is on disk.

Cannot Open COMMAND.COM

The source drive does not have COMMAND.COM. Use the COPY command to copy COMMAND.COM from a bootable volume onto the source drive.

Cannot Create MSDOS.SYS

-- or --

Cannot Create COMMAND.COM

There is no room left in the directory of the destination drive. Delete some files and try again.

No space for MSDOS.SYS

There is no room on the disk for MSDOS.SYS (the system image). Delete some files and try again. Note that MSDOS.SYS must reside on contiguous disk space. Thus if the diskette or volume is "checker boarded" (i.e., small, noncontiguous disk spaces available) you may have to delete many files. In this case, it is probably best to start with a newly formatted diskette or a completely empty volume; first do the SYSCOPY and then use the COPY command to copy the rest of the files.

No space for COMMAND.COM

There is no room on the destination disk for COMMAND.COM. Check the size of COMMAND, using the DIR command, and remove enough files from the destination drive to make room.

8.7.5 UNEXPECTED TERMINATION

This is applicable only if the destination drive is a hard disk volume. In order to prevent leaving the hard disk in an inconsistent state, when a copy is started whose destination is a hard disk volume, the system information which points to the boot image is cleared. This information is reinstated when the copy is completed. Thus, if a failure occurs during a copy to a hard disk volume (i.e., while the message "Copying..." is being displayed), you will not be able to boot from the hard disk. You will have to boot from the floppy. This condition will be corrected after the next successful SYSCOPY or assignment in HDSETUP. It is therefore very important that you keep a copy of the hard disk operating system, with its utilities, in a safe and handy place.

8.8 THE SDCOPY UTILITY

8.8.1 COMMAND FORMAT AND FUNCTION

The command format for the SDCOPY utility is -

```
SDCOPY [drivename:](cr)
```

where drivename: specifies the floppy disk drive to be used to make the copy.

SDCOPY copies the contents of one diskette (the source diskette) onto another diskette (the destination diskette). THE DESTINATION DISKETTE MUST BE FORMATTED.

You can use SDCOPY with single- or double-sided diskettes. If you use SDCOPY to copy a single-sided diskette onto a double-sided diskette, it transforms the double-sided diskette into a single-sided diskette.

8.8.2 COMMAND OPERATION

When you load SDCOPY (by entering the SDCOPY command), the utility displays its one-line sign-on banner. It then prompts you to insert your source and destination diskettes, in order. Multiple insertions are necessary, depending on memory size. SDCOPY reads as many sectors from the source diskette as it can hold in memory, prompts you to insert the destination diskette, writes the sectors in memory to the destination diskette - and repeats the process until the copy is complete.

You can exit the utility by pressing the Return key in response to a prompt for a diskette. If you exit the utility before the copy is complete, the destination diskette is unusable.

8.8.8 Command Example

The following example illustrates using SDCOPY on floppy drive B.

```
     B>sdcopy(cr)

Single Drive Disk Copy Utility, Ver. 1.Ø
```

```
Insert source disk, then press space bar (return key
to exit) (space)

3 insertions of your source and destination disks
will be required.

Partial read #1 of 3

Insert formatted destination disk, then press space bar
(return key to exit) (space)

Partial write #1 of 3

Insert source disk, then press space bar (return key to
exit) (space)

Partial read #2 of 3

Insert formatted destination disk, then press space bar
(return key to exit) (space)

Partial write #2 of 3

Insert source disk, then press space bar (return key to
exit) (space)

Partial read #3 of 3

Insert formatted destination disk, then press space bar
(return key to exit) (space)

Partial write #3 of 3

SDCOPY completed

Insert source disk, then press space bar (return key to
exit) (cr)
B>
```

9. MS-DOS INTERNAL COMMANDS

9.1 INTRODUCTION

The MS-DOS internal commands are presented in Table 9-1 and then described in detail.

Table 9-1: MS-DOS Internal Commands

COMMAND NAME	COMMAND FUNCTION
COPY	Copies specified file(s).
DATE	Sets and displays date.
DEL	Deletes specified file(s).
DIR	Lists specified directory entries.
PAUSE	Pauses for input in a batch file.
REM	Displays a comment in a batch file.
REN	Renames a file.
TIME	Sets and displays the time.
TOD	Displays the time and date.
TYPE	Displays the contents of the specified file.
?	Displays the current System drive and COMMAND version number.

9.2 THE COPY COMMAND

9.2.1 COMMAND FORMAT AND FUNCTION

The command format for COPY is --

```
COPY filespec[+filespec...][/A|/B]
[filespec][/A|/B](cr)
```

The COPY function is to copy the first file (or files) specified onto the second file specified.

9.2.2 COMMAND OPERATION

File Specifier Parameters

If the second file specifier is not given, the copy is on the default drive and has the same name as the file copied.

If the first file specifier is on the default drive and the second file specifier is not given, the copy is aborted. (Copying files to themselves is not allowed.) MS-DOS returns the error message:

```
File cannot be copied onto itself
Ø File(s) copied
```

The second file specifier may take three forms:

- o A drive designation only causes the original file to be copied with the same name to the designated drive.

- o A file name only causes the original file to be copied to a file with the name specified on the default drive.

- o A full file specifier cause the original file to be copied to a file with the name specified on the designated drive.

File Concatenation

You can concatenate files by simply listing any number of files as parameters to COPY, separating them by a plus sign (+).

For example, the command --

```
     COPY A.XYZ+B.COM+B:C.TXT BIGFILE.CRP(cr)
```

would concatenate the contents of A.XYZ, B.COM, and B:C.TXT and place them in the file on the default drive called BIGFILE.CRP.

The /A and /B Switches

Concatenation is normally carried out in text (or ASCII) mode, which means that an ALT Z (1A hex) in the file is interpreted as the end-of-file mark. To combine binary files, use the /B switch, which forces the command to use the physical end-of-file as the end of file (that is, the file length seen in the DIR command).

For example, the command --

```
     COPY/B A.COM + B.COM(cr)
```

would concatenate the binary files A.COM and B.COM (since a second file specifier is not given with the command, the resulting file would be named A.COM).

ASCII and binary files may be arbitrarily combined by using /B on binary files and /A on ASCII files. A /A or /B switch applies to the file it is placed after and to all subsequent files until the next /A or /B switch.

A /A or /B switch on the destination file determines whether or not an ALT Z is placed at the end of the file. (Source files that are read while /A is in effect have ALT Z stripped off. If /A is in effect when the file is written, a single ALT Z is put back.) Thus, an additional ALT Z would be appended with a command such as --

```
COPY  A.ASM/B B.ASM/A(cr)
```

The /B on the first file specifier in this command prevents the ALT Z from being stripped, and the /A on the second file specifier inserts an ALT Z. The primary practical application may be the reverse, where an ALT Z is stripped from the file with a command such as --

```
COPY PROG.COM/B + ERRS.TXT/A NEWPROG.COM/B(cr)
```

In this example, the file ERRS.TXT, which was generated by an editor, is actually considered constant data (error messages) by the program it is being appended to. Since the result is a .COM file, an ALT Z at the end is not needed.

The /A and /B switches are processed whether or not files are concatenated. When not concatenating, the COPY command defaults to binary copy. If the /A switch is used, the resulting file may end at the first end-of-file mark -- for example with a command such as --

```
COPY A.TXT/A B.TXT(cr)
```

If the file A.TXT in the preceding command contained an embedded ALT Z, the resulting file, B.TXT, would be shorter that A.TXT and would contain one ALT Z, the last character of the file.

Concatenation with Ambiguous File Names

You can also concatenate file names that contain wild-card characters. For example, the command --

```
COPY *.LST COMBIN.PRN(cr)
```

would combine all files matching *.LST into one file called COMBIN.PRN.

Several individual concatenations can also be performed. For example, the command --

```
COPY  *.LST + *.REF *.PRN(cr)
```

would combine each file that matched *.LST with the corresponding .REF file. The resulting files retain their original file names, but are given the extension .PRN. For example, in the preceding command, FILE1.LST would be combined with FILE1.REF to form FILE1.PRN, XYZ.LST would be combined with XYZ.REF to form XYZ.PRN, and so on. The command --

```
COPY *.LST + *.REF COMBIN.PRN(cr)
```

would combine all files matching *.LST with all files matching *.REF into one file named COMBIN.PRN.

Be careful not to enter a concatenation COPY command where one of the source files has the same name as the destination file. As each input file is found, its name is compared with the destination. If they are the same, that one input file is skipped, and the message --

```
Content of destination lost before copy
```

is printed. Further concatenation proceeds normally. This allows "summing" files, with a command like --

```
COPY ALL.LST + *.LST(cr)
```

This command appends all *.LST files, except ALL.LST itself, to ALL.LST. The error message is suppressed in this case, since this is produced by a true physical append to ALL.LST.

9.3 THE DATE COMMAND

Format and Function:

The format for the DATE command is --

DATE [dd-mm-yy] (cr)
or
DATE [mm/dd/yy] (cr)

where mm represents the number of the month, dd represents the number of the day, and yy represents the number of the year. (One-digit numbers may be entered for the number of the month and the number of the day.)

The function of the DATE command is to set and display the date.

Operation:

If entered without a parameter, the DATE command displays the message --

```
Current date is Day  mm/dd/yy or dd-mm-19yy
Enter new date:
```

where Day represents the name of the day, mm represents the number of the month, dd represents the number of the day, and yy represents the number of the year.

Press the Return key if you do not want to change the date shown.

Optionally, the date may be given as a parameter to the DATE command, in which case no message is displayed.

Use only numbers to enter the new date. The allowable parameters are --

- o 1-12 for the number of the month.
- o 1-31 for the number of the day.
- o 8Ø-99 or 198Ø-2Ø99 for the number of the year.

Separate the date, month, and year entries by hyphens for European format or slashes (/) for American format. MS-DOS is programmed to change months and years correctly, whether the month has 31, 3Ø, 29, or 28 days.

If the parameters or separators entered are not legal, MS-DOS displays the message --

```
Invalid date
Enter new date:
```

To solve this problem, enter a new date using legal parameters.

9.4 THE DEL COMMAND

The format for the DEL command is --

```
DEL filespec(cr)
```

The function of the DEL command is to delete all files that match the specified file specifier.

Wild-card characters may be used in the file specification parameter.

If the file name given is *.*, MS-DOS displays the prompt "Are you sure?" If you type a Y in response, all files are deleted.

9.5 THE DIR COMMAND

Format and Function:

The format for the DIR command is --

DIR [drivename:][filename[.ext]][/P][/W](cr)

The function of the DIR command is to list the files in a diskette's file directory.

Operation:

- o Without parameters, DIR lists all directory entries on the default drive.
- o With only a drive specification, DIR lists all entries on the diskette in the specified drive.
- o With only a file name, DIR lists all files with the specified file name on the diskette in the default drive.
- o With a full file specification, DIR lists all files that match that file specification.

In all cases, files are listed with their size in bytes and the time and date of their last modification.

Wild-card characters may be used in the file name and file extension parameters.

Two switches may be used with DIR. The /P switch selects page mode. With /P, display of the directory pauses after the screen is filled. To resume display of output, type any key.
The /W switch selects wide display. With /W, only file names are displayed (other file information is suppressed). File names are displayed five per line.

9.6 THE PAUSE COMMAND

Format and Function:

The format for the PAUSE command is --

PAUSE [comment](cr)

The function of the PAUSE command is to suspend execution of a batch file.

Operation:

Use PAUSE to allow time for necessary action -- such as changing a diskette -- during execution of a batch file. When COMMAND encounters PAUSE, it displays the optional comment and the message --

Strike any key when ready . . .

Pressing any key except ALT C resumes execution of the batch file. If you type an ALT C, COMMAND displays the prompt --

Abort batch job (Y/N)?

If you type Y, execution of the remainder of the batch command file is aborted, and control returns to the operating system. Therefore, PAUSE can be used to break a batch file into pieces, allowing you to end the batch command file at an intermediate point.

9.7 THE REM COMMAND

The format for the REM command is --

REM comment(cr)

The function of the REM command is to display a comment during execution of a batch file.

The only delimiters for the comment are any one of the three legal delimiters to start the comment (blank space, tab, or comma).

9.8 THE REN COMMAND

Format and Function:

The format for the REN command is --

```
REN [drivename:]filename[.ext]
filename[.ext](cr)
```

The function of the REN command is to rename a file or files.

Operation:

The first file specifier must designate a drive if the file resides on a diskette in a drive other than the default drive. (Any drive designation given for the second file specifier is ignored. The file remains on the diskette where it currently resides.)

Wild-card characters may be used in either parameter. A wild-card character causes a one-to-one correspondence between files specified by the first parameter and those specified by the second parameter.

For example, the command --

```
REN *.LST *.PRN(cr)
```

would change the extension of all files with the extension to .PRN.

As another example, the command --

```
REN B:ABODE ?D?B?(cr)
```

would rename the file ABODE on drive B to ADOBE (the file would remain on drive B).

9.9 THE TIME COMMAND

Format and Function:

The format for the TIME command is --

```
TIME [hh:mm[:ss]](cr)
```

where hh represents the number of the hour, mm represents the number of the minute, and ss represents the number of the second. (One-digit numbers may be entered for the numbers of the hour, minute, and second.)

The function of the TIME command is to set and display the time.

Operation:

If TIME is entered without parameters, MS-DOS displays the message --

```
Current time is hh:mm:ss.cc
Enter new time:
```

where hh represents the number of the hour, mm represents the number of the minute, ss represents the number of the second, and cc represents the number of the half-second.

Press the Return key to leave the time set as shown. If TIME is entered with parameters, no message appears.

Use numbers to enter the new time. The allowable parameters are --

- ØØ-23 for the number of the hour.
- ØØ-59 for the number of the minute.
- ØØ-59 for the number of the second.

The hour, minute, and second entries must be separated by colons (:). If the parameters entered are not legal, MS-DOS displays the message --

Invalid time
Enter new time:

To solve this problem, enter a new time using legal parameters.

9.1Ø THE TOD COMMAND

The TOD command displays the current date and time.

9.11 THE TYPE COMMAND

The format for the TYPE command is --

TYPE filespec(cr)

The function of the TYPE command is to display the contents of the specified file on the console screen.

The only formatting performed by TYPE is that tabs are expanded to spaces consistent with tab stops every eighth column. Note that display of binary files causes alternate characters to be sent to your computer, including bells, form feeds, and escape sequences.

9.12 SYSTEM AND USER DRIVES

This feature allows the user to make distinct the default (user) drive and the drive (system) from which commands are obtained. The user drive may be changed without disturbing the system drive, allowing the user to access data without losing the ability to group programs and batch files on a single drive. The syntax follows COMMAND standard option specification.

a: -- change the default drive to a: and system drive to the default drive.

d:/u -- change only the default drive to a:, leaving the system drive unchanged.

c:/s -- change the system drive but leave the default drive unchanged.

e:/s/u is equivalent to e:/u.

The ? command will display the current system drive, if it is not the default drive.

10. EDLIN: THE MS-DOS TEXT EDITOR

EDLIN is a text editor used to edit files that are divided into lines. Each line may contain up to 255 characters. The last character of each line is the end-of-line character, the carriage return.

10.1 EDLIN LINE NUMBERS

When EDLIN displays a file, it numbers the lines dynamically, beginning with line number 1. (Line numbers are not actually present in saved text.) When new lines are inserted or existing lines deleted, line numbers are adjusted accordingly. Consequently, line numbers always run from 1 through the number of the last line.

10.2 EDLIN OVERVIEW

10.2.1 EDITING AN EXISTING FILE

To invoke EDLIN to edit an existing file, enter --

EDLIN filespec(cr)

where filespec is the file specifier of the file you wish to edit. If the file exists, EDLIN loads the file into memory. If EDLIN is able to load the whole file, the system displays the message --

End of input file

and the EDLIN asterisk (*) command prompt. If EDLIN is not able to load the whole file, it fills 3/4 or available memory with the first part of the file and does not display the "End of input file" message.

To edit the part of a file that is not in memory, you write out to disk some of the file that is in memory, and then append lines into memory.

The commands for writing to memory and for appending additional text are --

[n]W(cr)

where n is the number of lines to be written out; and --

[n]A(cr)

where n is the number of lines to be appended.

When the editing session ends, you save the file on the same drive where it was found by typing the command --

E(cr)

10.2.2 CREATING A NEW FILE

To create a new file, invoke EDLIN by entering --

EDLIN filespec(cr)

where filespec is the name you wish to give to the new file. EDLIN creates the file and displays the message --

NEW FILE

It then displays the asterisk (*) command prompt. To begin the editing session enter an I command. When you are finished entering data, enter an ALT-C or ALT-Z. To display the file you have created, enter an L command. The I, L, and all other editing commands are described later in this chapter.

The commands for writing data to memory and ending the editing session are the same as those used for editing existing files.

IMPORTANT: When creating a new file, be sure to specify on which drive the file should be saved. Otherwise, the file will be saved on the default drive.

1Ø.2.3 EDLIN EDITING FUNCTIONS AND COMMANDS

EDLIN offers two types of editing functions: intraline functions and interline commands. Intraline functions are used to edit a single line; interline commands are used to manipulate whole lines of text.

1Ø.3. INTRALINE FUNCTIONS

The intraline functions are the Alternate-character functions and the special editing functions discussed in Chapter 7. The only difference is that the EDLIN functions operate on the lines of an EDLIN file, rather than the MS-DOS command line.

Only the special editing functions are described in this chapter. See 7.5 for more information on the Alternate-character functions.

Table 1Ø-1 summarizes the EDLIN special editing functions. These functions are accomplished by entering the escape sequences listed in Table 1Ø-1 (plus the specified character, if necessary). The function keys on your keyboard can also be programmed to perform the special editing functions. Your dealer can provide you with a system diskette that allows such use of the function keys.

Use of the special editing functions is described following Table 1Ø-1.

Table 10-1: Special Editing Functions

FUNCTION NAME	ESCAPE SEQUENCE*	EDITING EFFECT
C1	Escape S	Copies one character from the template to the command line.
CM	Escape T	Copies all characters up to the specified character from the template to the command line.
CT	Escape U	Copies all remaining characters in the template to the command line.
S1	Escape V	Skips over (does not copy) a character in the template.
SM	Escape W	Skips over (does not copy) the characters in the template up to the specified character.
QI	Escape H	Voids the current input; leaves the template unchanged.
INS	Escape P	Enters insert mode.
REP	Escape Q	Exits insert mode (toggle from insert); the REP mode is the default mode.

Table 10-1 (continued)

FUNCTION NAME	ESCAPE SEQUENCE*	EDITING EFFECT
NT	Escape R	Makes the new line the new template.

* The escape sequences listed in this column are accomplished by pressing the Esc key and then pressing the specified letter key. The letter keys must be typed in upper case.

10.3.1 THE C1 FUNCTION

The C1 function is to copy one character from the template to the input buffer.

When the C1 function is used, one character is inserted in the buffer and insert mode is automatically turned off if it was on. Use the C1 function to advance the cursor one column across the line. For example, assume the screen shows:

```
      1:*This is a sample file.
      1:*_
```

Entering Escape S copies the first character (T):

```
      1:*This is a sample file
ESC-S 1:*T_
```

Each time Escape S is entered, one more character appears:

```
ESC-S 1:*T
ESC-S 1:*Thi
ESC-S 1:*This_
```

10.3.2 THE CM FUNCTION

The CM function is to copy multiple characters up to a given character from the template to the input buffer.

The specified character is the next character typed after the CM function is used. Using the CM function causes the cursor to move to the specified character. If the template does not contain the specified character, nothing is copied. The CM function automatically turns off insert mode if it is on. For example, assume the screen shows:

```
                1:*This is a sample file.
                1:*_
```

Entering Escape T copies all characters up to the character pressed immediately after the Escape T.

```
                1:*This is a sample file
        ESC-T p 1:*This is a sam_
```

10.3.3 THE CT FUNCTION

The CT function is to copy the template to the input buffer.

Regardless of the cursor position at the time the CT function is used, the whole line is displayed, and the cursor is positioned after the last character on the line. The CT function automatically turns off insert mode if it was on. For example, assume the screen shows:

```
                1:*This is a sample file.
                1:*_
```

Entering Escape U copies all characters from the template (shown in the upper line displayed) to the line with the cursor:

```
              1:*This is a sample file
        ESC-U 1:*This is a sample file._
```

10.3.4 THE S1 FUNCTION

The S1 function is to skip over one character in the template.

Each time you use the S1 function, one character is deleted (not copied) from the template. The S1 function is similar to the C1 function, except that S1 skips a character in the template rather than copies it to the input buffer. For example, assume the screen shows:

```
              1:*This is a sample file.
              1:*_
```

Entering Escape V skips over the first character (T):

```
              1:*This is a sample file
        ESC-V 1:*_
```

The cursor position does not move; only the template is affected. To see how much of the line has been skipped over, enter Escape U (the CT function, which moves the cursor beyond the last character of the line).

```
              1:*This is a sample file.
        ESC-V 1:*_
        ESC-U 1:*his is a sample file._
```

10.3.5 THE SM FUNCTION

The SM function is to skip over multiple characters in the template, up to a specified character.

If the template does not contain the specified character, nothing is skipped over. The SM function is similar to the CM function, except that SM skips over characters in the template rather than copies them to the input buffer. For example, assume the screen shows:

```
                1:*This is a sample file.
                1:*_
```

Entering Escape W skips over (deletes) all the characters in the template up to the next character pressed.

```
                1:*This is a sample file
          SC-W p1:*_
```

The cursor position does not move. To see how much of the line has been skipped over, enter Escape U to copy the template. This moves the cursor beyond the last character of the line:

```
                1:*This is a sample file:
         ESC-W p 1:*_
         ESC-U   1:*ple file._
```

10.3.6 THE QI FUNCTION

The QI function is to quit input and flush the input buffer (the template is unchanged).

The QI function also prints a backslash (\), carriage return, and line feed, and turns insert mode off if it was on. The cursor is positioned at the beginning of the line. Using CT copies the template to the input buffer just as the line was before QI was used.

Assume the screen shows:

```
            1: This is a sample file.
            1:*_
```

Assume you want to replace the line with the new line "Sample File."

```
            1:*This is a sample file.
Sample File 1:*Sample File_
```

Now, to reedit the line, enter Escape H:

```
            1:*This is a sample file.
      ESC-H 1:*Sample File\
            1:*_
```

You can press the Return key to keep the original line or to perform any other intraline editing function. If you enter Escape U (the CT function), the original template is copied to the input buffer:

```
      ESC-U 1:*This is a sample file._
```

10.3.7 THE INS FUNCTION

The INS function is to enter insert mode (the cursor position is not changed).

The cursor moves to the right as each character is inserted. Characters are inserted before the character the cursor points to; when you have finished inserting characters, the cursor is positioned at the same character as before the insertion began.

Assume the screen shows:

```
        1:*This is a sample file.
        1:*_
```

If you entered Escape T (the CM function) and pressed the character p, the result would be --

```
        1: This is a sample file
ESC-T p 1:*This is a sam_
```

If you now enter Escape P and insert the three characters s, o, and n -

```
          1:*This is a sample file.
ESC-T p   1:*This is a sam
ESC-P son 1:*This is a samson_
```

and then entered Escape U (the CT function), the rest of the template would be copied to the line --

```
          1:*This is a samson
    ESC-U 1:*This is a samsonple file._
```

If you were to press the Return key instead of entering Escape U, the rest of the template would be deleted, and the input buffer ended at the end of the insert --

```
ESC-P son(cr) 1:*This is a samson
```

10.3.8 THE REP FUNCTION

The REP function is to exit from insert mode (if necessary) and enter replace mode. While in replace mode, all characters entered replace characters in the template.

Replace mode -- rather than insert mode -- is the default mode. When you start to edit a line, replace mode is in effect. If you press the Return key while in replace mode, the rest of the template is discarded.

Assume the screen shows:

```
                  1:*This is a sample file.
                  1:*_
```

Assume you enter Escape T and the character p (the CM function), Escape P and the characters son (the INS function), Escape Q and the characters ite (the REP function), and Escape U (the CT function) -

```
                  1:*This is a sample file.
        ESC-T p   1:*This is a sam
        ESC-P son 1:*This is a samson
        ESC-Q ite 1:*This is a samsonite
        ESC-U     1:*This is a samsonite file._
```

If you type in characters that extend beyond the length of the template, the remaining characters in the template are automatically appended when you enter Escape U.

1Ø.3.9 THE NT FUNCTION

The NT function creates a new template by copying the current contents of the input buffer to the template. The contents of the old template are destroyed.

The NT function outputs an at-sign character (@), a carriage return, and a line feed. The input buffer is emptied, and insert mode is turned off.

NOTE: NT performs the same function as QI, except that the template is changed, and an at-sign character (@) is printed instead of a backslash (\).

Assume the screen shows:

```
1:*This is a sample file.
1:*_
```

Assume that you enter Escape T and the character p (the CM function), Escape P and the characters son (the INS function), Escape Q and the characters ite (the REP function), and Escape U (the CT function) -

```
          1:*This is a sample file.
ESC-T p   1:*This is a sam
ESC-P son 1:*This is a samson
ESC-Q ite 1:*This is a samsonite
ESC-U     1:*This is a samsonite file._
```

At this point, if you wanted this line as the new template, you would enter Escape R --

```
ESC-R 1:*This is a samsonite file.@
```

You could now edit the new template.

1Ø.4 INTERLINE COMMANDS

Interline commands manipulate whole lines, rather than parts of lines. The interline commands are summarized in Table 1Ø-2 and described in detail in the following sections.

Table 10-2: EDLIN Interline Commands

COMMAND NAME	COMMAND FUNCTION
n*	Edit line
A	Append lines
D	Delete lines
E	End editing
I	Insert text
L	List text
Q	Quit editing
R	Replace text
S	Search text
W	Write lines

* In this entry n represents the number of the line to be edited.

10.4.1 INTERLINE COMMAND PARAMETERS

The parameters used with the interline commands are described in Table 10-3. The effect of a parameter depends on the command with which it is used.

Table 10-3: Interline Command Parameters

PARAMETER — DEFINITION

line number — A line-number parameter must be separated from other parameters and the command by a comma or a blank space. The line number may be specified by entering one of the following:

o **A number** that is less than 65534 (a file cannot contain more than 65534 lines). If a number larger than the largest existing line number is specified, the line-number parameter indicates the line after the last existing line.

o **A period (.)**, which indicates the current line number. The current line is the last line edited (not necessarily the last line displayed). The current line is marked on the screen by an asterisk (*), which is placed between the line number and the first character of the line.

o **A pound sign (#)**, which indicates the line after the last line number. Specifying # for the line-number parameter has the same effect as specifying a number larger than the last line number.

Table 1Ø-3 (continued)

PARAMETER	DEFINITION
	o **A Return** (without a number, period, or pound sign), which assigns a line-number default value appropriate to the command.
?	The question mark parameter directs EDLIN to ask if the correct string has been found. The question mark is used only with S and R commands (search and replace). Before continuing, EDLIN waits for a Y or a Return for a yes response, or for any other key for a no response.
string	(where string represents a string of text to be found, to be replaced, or to replace other text.) The string parameter is used only with the S and R commands (search and replace). Each string parameter must end with an ALT-Z or a Return (see the R command for details). No spaces should be left between string parameters or between a string parameter and its command letter, unless the spaces are meant to be part of the string.

1Ø.4.2 THE EDIT COMMAND

Format and Function:

The EDIT command format is --

[n|.](cr)

where n represents a line number; see Table 1Ø-3 for an explanation of the line number parameters.

Its function is to prepare EDLIN to edit the specified line.

Operation:

When a line number is specified with one of the line-number parameters, EDLIN displays the line number and the line text, and then, on the line below, it displays the line number and asterisk prompt. The line is then ready for editing. You may use any of the intraline commands to edit the line. The existing text of the line serves as the template until you press the Return key.

If the Return key only is pressed, EDLIN prepares to edit the line after the current line. If no changes of the current line are needed and the cursor position is at the beginning or end of the line, press the Return key to accept the line as is.

WARNING: If the Return key is pressed while the cursor is in the middle of the line, the rest of the line is deleted.

Assume the following file exists and is ready to edit:

```
1: This is a sample file.
2: used to demonstrate
3: the editing of line
4:*four.
```

To edit line 4, you would enter the number 4. The contents of the line are displayed along with the cursor below the line:

```
4:*four.
4:*_
```

If you wanted to insert the word "number" in line 4, you would type --

```
ESC-P number 4:*number
ESC-U(cr)    4:*number four.
```

10.4.3 THE A COMMAND (APPENDING LINES OF TEXT)

The A command format is --

[n]A(cr)

Its function is to append line(s) from the input file to the editing buffer.

Use this command for extremely large files that do not fit into memory all at one time. By writing out part of the editing buffer to the output file with the W command, room is made for lines to be appended with the A command.

If a number is entered with the A command, that number of lines is appended to the part of the file currently in memory. If a number is not entered, the input file is read into the editing buffer until the editing buffer is 3/4 full.

10.4.4 THE D COMMAND(DELETING LINES OF TEXT)

Format and Function:

The D command format is --

[n|,n|,#|n,n|n,#] D(cr)

Its function is to delete the specified line or lines.

Operation:

The D command operates as follows:

- o Without parameters, it deletes the current line.
- o With one line number, it deletes the specified line.
- o With one line number (or the pound sign) preceded by a comma, it deletes the current line, the specified line, and all lines between the current line and the specified line.
- o With a line number followed by a comma and another line number or the pound sign, it deletes the specified lines and all lines between them.

When lines have been deleted, the line immediately after the deleted lines becomes the current line.

Assume the following file exists and is ready to edit:

```
 1: This is a sample file.
 2: Use: to demonstrate dynamic line numbers
 3: See what happens when you
 4: Delete and Insert
  .
  .
  .
25: (The D and I commands)
26: (Use ALT-C to exit insert mode)
27:*Line numbers
```

To delete lines 5 through 24, you would enter --

```
5,24 D(cr)
```

The result would be --

```
1: This is a sample file.
2: Use: to demonstrate dynamic line numbers
3: See what happens when you
4: Delete and Insert
5:*(The D and I commands)
6: (Use ALT-C to exit insert mode)
7: Line numbers
```

To delete line 6, you would enter --

```
6 D(cr)
```

The result would be --

```
1: This is a sample file.
2: Use: to demonstrate dynamic line numbers
3: See what happens when you
4: Delete and Insert
5: (The D and I commands)
6:*Line numbers
```

If you then added the words "delete and insert" to line 3, resulting in the following --

```
1: This is a sample file.
2: Use: to demonstrate dynamic line numbers
3: See what happens when you delete and insert
4:*Delete and Insert
5: (The D and I commands)
6: Line numbers
```

and then decided to delete lines 4 (the current line) and 5, you would enter --

```
,5 D(cr)
```

The result would be --

```
1: This is a sample file.
2: Use: to demonstrate dynamic line numbers
3: See what happens when you delete and insert
4:*Line numbers
```

1Ø.4.5 THE E COMMAND (ENDING AN EDLIN SESSION)

The E command format is --

```
E(cr)
```

Its function is to end the editing session.

The E command saves the edited file on disk, renames the original file's extension with the .BAK extension, and exits to the MS-DOS command level. If the file was created during the editing session, no .BAK file is created.

The E command takes no parameters. Therefore, you cannot tell EDLIN on which drive to save the file. If you did not designate the drive when you invoked EDLIN, the file is saved on the disk in the default drive.

Be sure that the disk contains enough free space for the entire file to be written. If the disk does not contain enough free space, the write is aborted, and the edited file lost.

10.4.6 THE I COMMAND (INSERTING LINES OF TEXT)

Format and Function:

The I command format is --

```
     [n|#] I(cr)
```

Its function is to insert line(s) of text immediately before the specified line.

Operation:

Enter the I command to create a new file. In this case, the insert begins with line number 1.

EDLIN remains in insert mode until you enter an ALT-Z or an ALT-C. Successive line numbers appear automatically each time you press the Return key. When you exit from insert mode, the specified line, which now immediately follows the inserted lines, becomes the current line. All line numbers following the inserted section are incremented by the number of lines inserted.

If a line number is not specified, the new lines are inserted immediately before the current line. If the specified line number is larger than the last line number, or if # is specified as the line number, the inserted lines are appended to the end of the file. In this case, the last line inserted becomes the current line.

Assume the following file exists and is ready to edit:

```
1: This is a sample file.
2: Use: to demonstrate dynamic line numbers
3: See what happens when you
4: Delete and Insert
5: (The D and I commands)
6: (Use ALT-Z or ALT-C to exit insert mode)
7:*Line numbers
```

To insert text before line 4, you would enter --

```
4 I(cr)
```

and the screen would display --

```
4:_
```

Assume that you now entered the new lines --

```
4: fool around with(cr)
5: those very useful commands that(cr)
```

and ended the insertion by entering an ALT-Z --

```
6:^Z
```

You could now list the new file with an L command by typing --

```
L(cr)
```

The result would be --

```
1: This is a sample file.
2: Use: to demonstrate dynamic line numbers
3: See what happens when you
4: fool around with
5: those very useful commands that
6:*Delete and Insert
7: (The D and I commands)
8: (Use ALT-Z or ALT-C to exit insert mode)
9: Line numbers
```

To insert lines immediately before the current line, you would enter --

```
I(cr)
```

and the screen would display --

```
6:*_
```

If you then entered the following line, ending it with an ALT-Z --

```
6:* perform the two major editing functions(cr)
7:*^Z
```

and typed an L command to list the file, the result would be --

```
1: This is a sample file.
2: Use: to demonstrate dynamic line numbers
3: See what happens when you
4: fool around with
5: those very useful commands that
6: perform the two major editing functions,
```

1Ø.4.7 THE L COMMAND (LISTING LINES OF TEXT)

Format and Function:

The L command format is --

```
[n,n|n,#|,n|,#|n,] L(cr)
```

The L command function is to list the specified line or range of lines, including the two lines specified.

Operation:

The L command operates as follows:

- With no parameter, it lists the 23 lines centered around the current line (the 11 lines preceding the current line, the current line, and the 11 lines following the current line).

- With two line numbers or a line number and the pound sign (separated by a comma), it lists the specified range of lines.

- With one line number or the pound sign preceded by a comma, it lists the range of lines from the current line to the specified line.

- With one line number followed by a comma, it lists a range of 23 lines centered around the specified line (the 11 lines preceding the specified line, the specified line, and the 23 lines following the specified line.

If the current line is one of the lines listed, it contains an asterisk between the line number and the first character.

Assume the following file exists and is ready to edit --

```
     1: This is a sample file.
     2: Use: to demonstrate dynamic line numbers
     3: See what happens when you
     4: Delete and Insert
     5: (The D and I commands)
         .
         .
         .
    15:*The current line contains an asterisk.
         .
         .
         .
    26: (Use ALT-Z or ALT-C to exit insert mode)
    27: Line numbers
```

To list the range of lines from line 2 to line 5, you would enter --

```
    2,5 L(cr)
```

The result would be --

```
    2: Use: to demonstrate dynamic line numbers
    3: See what happens when you
    4: Delete and Insert
    5: (The D and I commands)
```

To list a range of lines from the current line to line 26, you would enter --

```
    ,26 L(cr)
```

The result would be --

```
15:*The current line contains an asterisk.
    .
    .
    .
26: (Use ALT-Z or ALT-C to exit insert mode)
```

To list a range of 23 lines around line 13, you would enter --

```
13, L(cr)
```

The result would be --

```
13: The specified line is listed first in the
    range.
14: The current line remains unchanged by the L
    command.
15:*The current line contains an asterisk.
    .
    .
    .
35: ALT-C exits interline insert command mode.
```

To list a range of 23 lines centered around the current line, you would enter --

```
L(cr)
```

The result would be --

```
     2: Use: to demonstrate dynamic line numbers
     3: See what happens when you
     4: Delete and Insert
     5: (The D and I commands)
        .
        .
        .
    13: The current line is listed in the middle of
        the range.
    14: The current line remains unchanged by the L
        command.
    15:*The current line contains an asterisk.
        .
        .
        .
    24: ALT-C exits interline insert command mode.
```

10.4.8 THE Q COMMAND (QUITTING AN EDLIN SESSION)

The Q command format is --

```
     Q(cr)
```

Its function is to quit the editing session (it does not save editing changes) and exit to the MS-DOS command level.

The Q command is simply a quick means to exit an editing session. When the Q command is given, EDLIN displays the message --

```
     Abort edit (Y/N)?
```

Press Y to quit the editing session; press N (or any other key except ALT-C) to continue the editing session.

10.4.9 THE R COMMAND (REPLACING STRINGS OF TEXT)

Format and Function:

The R command format is --

[n|n,n|,n|.,n|n,.] [?] Rstring[ALT-Zstring](cr)

Its function is to delete all occurrences of a string of text within a specified range, or replace the string with a second string of text.

Operation:

The R command operates in the following manner:

o String parameters:

- If only one string is specified, the R command deletes all occurrences of the string within the specified range.

- If two strings (separated by an ALT-Z) are specified, the R command replaces the first string with the second string within the specified range.

o Range parameters:

- No parameter specifies the range from the line after the current line to the end of the file.

- n specifies the range from the specified line to the end of the file (i.e., the range for n equals the range for n,#).

- n,n specifies the range from the first specified line through the second specified line.

  - ,n specifies the range from the line after the current line through the specified line.

  - .,n specifies the range from the current line through the specified line.

  - n,. specifies the range from the specified line through the current line.

- The ? parameter causes EDLIN to prompt the user to OK each string replacement or deletion.

  - When EDLIN encounters a matching string of text, it displays the line and the prompt "O.K.?".

  - If you press Y or the Return key, EDLIN makes the replacement or deletion and searches for the next matching string.

  - If you press any key other than Y or the Return key, EDLIN leaves the string as is and searches for the next matching string.

Assume the following file exists and is ready for editing:

```
 1: This is a sample file.
 2: Use: to demonstrate dynamic line numbers
 3: See what happens when you
 4: fool around with
 5: those very useful commands that
 6: perform the two major editing functions,
 7:*Delete and Insert
 8: (The D and I commands)
 9: (Use ALT-C to exit insert mode)
10: Line numbers
11: The insert command can place new lines
12: anywhere in the file; there's no space
    problems.
13: because the line numbers are dynamic;
```

```
    14: They'll slide all the way to 65533.
```

To replace all occurrences of "and" with "or" in or between lines 2-12, you would enter --

```
     2,12 Rand^Zor(cr)
```

This would result in the following changes:

```
     5: those very useful commands that
     7: Delete or Insert
     8: (The D or I commors)
    11: The insert commor can place new lines
```

To avoid the unwanted substitutions, you could have entered the command --

```
     12 ? Rand^Zor(cr)
```

If you then confirmed only those changes that made grammatical sense, the following would be displayed on your screen:

```
     5: those very useful commors that
     O.K.? N
     7: Delete or Insert
     O.K.? Y
     8: (The D or I commors)
     O.K.? N
    11: The insert commor can place new lines
     O.K.? N
     *
```

If you then entered an L command to list the file, the changed lines would be as follows --

```
     .
     .
  5: those very useful commands that
     .
  7: Delete or Insert
  8: (The D or I commands)
     .
 11: The insert command can place new lines
     .
     .
```

10.4.10 THE S COMMAND (SEARCHING FOR A STRING OF TEXT)

Format and Function:

The S command format is --

```
[n|n,n|,n|.,n|n,.] [?] Sstring(cr)
```

Its function is to search the specified range of lines for specified string and display the first line containing the string. This line (or, if the ? parameter is given, the first line that contains the string and is approved by the user) becomes the current line.

Operation:

The S command operates as follows:

o Range parameters:

  - No parameter specifies the range from the line after the current line to the end of the file.

  - n specifies the range from the specified line to the end of the file (i.e., the range for n equals the range for n,#).

  - n,n specifies the range from the first specified line through the second specified line.

  - ,n specifies the range from the line after the current line through the specified line.

  - .,n specifies the range from the current line through the specified line.

  - n,. specifies the range from the specified line through the current line.

- The ? parameter causes EDLIN to prompt you to OK the displayed line as the current line.

  - When EDLIN encounters a matching string of text, it displays the line and the prompt "O.K.?".

  - If you press Y or the Return key, the displayed line becomes the current line.

  - If you pres any key other than Y or the Return key, EDLIN searches for the next matching string.

Assume the following file exists and is ready for editing:

```
1: This is a sample file.
2: Use: to demonstrate dynamic line numbers
3: See what happens when you
4: fool around with
5: those very useful commands that
```

```
 6: perform the two major editing functions,
 7: Delete and Insert
 8: (The D and I commands)
 9: (Use ALT-C to exit insert mode)
10: Line numbers
11: The insert command can place new lines
12: anywhere in the file; there's no space
    problems.
13: because the line numbers are dynamic;
14:*They'll slide all the way to 65533.
```

To search for the first occurrence of the word "and" within the range of lines 2-14, enter --

```
2,12 Sand(cr)
```

The result is --

```
5: those very useful commands that
```

To get the "and" in line 7, modify the search command by entering --

```
,12 Sand(cr)
```

The search then continues from the line after the current line (line 5), since no first line is given. The result is:

```
7: Delete and Insert
```

To search the file for the first occurrence of the word "and," you would enter the command --

```
1 ? Sand(cr)
```

If you then OKed the first line that contained the word "and" as the current line, the resulting display would be --

```
5: those very useful commands that
O.K.? N
7 Delete and Insert
O.K.?Y
*
```

1Ø.4.11 THE W COMMAND (WRITING LINES TO MEMORY)

The W command format is --

```
[n]W(cr)
```

Its function is to write lines from the editing buffer to the output file.

Use the W command when editing files that are larger than available memory. The W command writes lines out to the output file, and makes room in the input buffer for more lines to be appended from the input file.

If no parameter is given, lines are written until memory is 1/4 full. If a number is given, then that number of lines is written out. As lines are written, subsequent lines in the editing buffer are renumbered beginning with one.

An A command following the W command appends lines to any remaining lines in the editing buffer.

SECTION III

THE CP/M-86 OPERATING SYSTEM

11. INTRODUCTION

The CP/M-86 operating system provides a way for you to communicate with your computer. CP/M-86 must be loaded for you to use the application programs that operate with CP/M-86. Like other operating systems, CP/M-86 manages the basic functions used by these programs.

Chapters 11 through 16 describe how to use CP/M-86 files, commands, and utilities. You can use CP/M-86 commands or utilities whenever you are using the operating system. You know you are using the operating system when you see the DOS command prompt — a drive letter name followed by >.

(Note: If you have a hard disk, refer to 8.4 for information on the CP/M-86 emulator.)

11.1 CP/M-86 CAPABILITIES

CP/M-86 operating system commands work with files created by an application program. The powerful CP/M-86 utilities can be used as extra capabilities (or as alternatives) for application program operations in managing disk files. With CP/M-86 you can perform the following tasks (plus more complex tasks described in this section):

o Display summarized information about CP/M-86 functions (HELP).

o Display and set time of day (TOD).

o Prepare diskettes for use on your computer (FORMAT).

o Copy the contents of one diskette onto another (DCOPY).

- Copy the operating system tracks from one diskette to another (BOOTCOPY).

- Check the size and read/write status of disk files (STAT).

- Display a diskette directory or desired parts of a directory (DIR and DIRS).

- Display the contents of a disk file on the screen (TYPE).

- Erase and rename files (ERA and REN).

- Copy or move files from one diskette to another (PIP).

- Output files to a peripheral device (such as a printer) (PIP).

- Execute a series of commands with one command (SUBMIT).

- Move files between subdirectories (USER and PIP).

11.2 SECTION CONVENTIONS

This section uses the following conventions:

- Examples are shown in boldface, and what you enter is underlined.

- In examples and command-format illustrations, nonprinting ASCII characters, such as carriage returns, appear as symbols. For instance, (cr) represents a carriage return; (space) represents a space.

- The ALT (Alternate function) key acts as the "Control" key on your computer. In examples in this manual, ALT-key sequences are represented with a caret (^) followed by the ALT character. For example, ALT-C appears as ^C. (Use of the ALT key is described in 12.3.)

- Command names, file names, and system prompts or responses embedded in text appear in upper case.

- When command formats and standard user entries appear in text for the first time, they are presented in boldfaced single lines.

- In command formats:

  - Braces ({}) indicate that one item in the enclosed group must be selected.

  - A vertical bar (|) separates alternatives.

  - Brackets ([]) indicate that the enclosed item(s) are optional.

  - An ellipsis in brackets ([...]) indicates that the preceding parameter can be repeated any number of times. A comma preceding the ellipsis ([,..]) indicates that each parameter must be separated from the others by a comma.

  - General forms (which represent specific entries to be supplied by the user -- such as file names, drive names, and device names) appear as single words, such as filename, drivename, and logical device.

- The term system diskette refers to any diskette that contains the CP/M-86 operating system.

The following example command format illustrates some of the conventions presented in this section.

```
{dir|dirs} [drivename:|filespecifier] (cr)
```

The command stem, DIR or DIRS, is enclosed in braces to indicate that one of these alternatives must be included in the command. The command tail elements DRIVENAME: and FILESPECIFIER are enclosed in brackets to indicate that they are mutually exclusive options.

12. CP/M-86 OVERVIEW

CP/M-86 is an operating system for your computer. The structure of the CP/M-86 file system allows dynamic allocation of file space and both sequential and random file access. Using this file system, many programs can be stored in both source and machine-executable form on one diskette.

12.1 COMPONENTS

There are four main components of CP/M-86: (1) the Basic Input/ Output System (BIOS); (2) the Basic Diskette Operating System (BDOS); (3) the Console Command Process (CCP); and (4) the Transient Program Area (TPA). Component functions follow:

- BIOS -- provides the primitive input/output interface to the diskette drives and input/output devices.
- BDOS -- controls the diskette drives and file system.
- CCP -- reads the keyboard and processes commands.
- TPA -- holds programs loaded from diskette.

12.2 INPUT/OUTPUT DEVICES

CP/M-86 performs its input and output based on the four "logical" devices described in Table 12-1. A logical device is the name of a device (e.g., LST:) that programmers or operators refer to when doing input or output processing. Special-purpose programs, called drivers, handle the transfer of data to and from these logical devices. These drivers interface with specific physical devices, which are actual input/output devices such as disk drives and line printers.

The operator can assign a physical device to a logical device name through the STAT command (as described in 14.4). For example, the operator can direct output to any of four physical devices by changing the physical-to-logical device association, as described in 14.4.8.

Table 12-1 describes the CP/M-86 logical and physical devices.

The names of the actual devices attached to your computer may or may not correspond to the physical device names listed in Table 12-1. For example, the device TTY: is actually an RS-232-C serial communications port to which a printer, plotter, or modem might be attached. The driver programs in the BIOS define the correspondence of physical device to actual device.

Table 12-1: Logical and Physical Devices

DEVICE TYPE/NAME	DESCRIPTION
Logical Devices	
CON:	Console device -- the principal interactive console which communicates with the operator. Typically, CON: is a device such as a CRT or teletype.
LST:	List device -- the principal listing device; usually a hard-copy device, such as a printer or teletype.
AXI:	Auxiliary input device.
AXO:	Auxiliary output device.

DEVICE TYPE/NAME	DESCRIPTION
Physical Devices	
TTY:	Serial output -- port A (teletype-style printer -- RS-232-C)
CRT:	Keyboard and cathode ray tube display
LPT:	Parallel port printer
UL1:	Serial printer -- port B (RS-232-C)
BAT:	Batch mode-reader (AXI:) as input; a printer (LST:) as output.
UC1:	External console (to be developed)
PTR:	High speed read (to be developed)
UR1:	(To be developed)
UR2:	(To be developed)
PTP:	High speed punch (to be developed)
UP1:	(To be developed)
UP2:	(To be developed)

12.3 BASIC OPERATING PRINCIPLES AND NOTES

This section (1) describes basic operating principles for using CP/M-86 commands and (2) notes certain general characteristics of the operating system. Some of the operating principles are expanded in Chapter 13.

- Once CP/M-86 is loaded (as described in 13.1), the console displays the letter name of the default drive, followed by the command prompt >. At the operating-system level, CP/M-86 always displays the name of the default drive with the system command prompt (e.g., A> or B>).

- You initiate CP/M-86 functions by entering commands at the keyboard. The CP/M-86 commands and utility programs you use to manipulate the data contained on diskettes are described in Chapters 14, 15, and 16.

- When CP/M-86 does not recognize a command or part of a command, the system repeats the first unrecognized portion of the command, followed by a question mark.

- The "default drive" is the currently selected drive, the drive to or from which CP/M-86 is recording or obtaining data. Commands act on the diskette in the default drive, unless you specify another drive name in the command. Also, commands load from the default drive unless another drive is specified.

- Commands you enter in response to the > prompt are referred to as "system-level" commands.

- All the commands described in this manual can be entered in upper or lower case. CP/M-86 automatically converts commands entered in lower case at the keyboard into upper case.

o Your ALT (Alternate-function) key has two functions:

  — It displays the symbol or invokes the function shown on the front face of another key.

  -- It acts as the Control key (for example, the function generally referred to as a Control C is an ALT-C on your computer and is accomplished by striking C while pressing the ALT key).

12.4 ENTERING AND MODIFYING CP/M-86 COMMANDS

You communicate with the operating system by entering CP/M-86 commands at the keyboard. Pressing the Return key at the end of a command line sends the command to the operating system. A command line can be up to 127 characters long.

You can modify CP/M-86 commands at any time during entry, until the Return key is pressed. To modify the command line, use the ALT-key functions described in Table 12-2.

Table 12-2: ALT-Key Command-Line Editing Functions

ALT CHARACTER	FUNCTION
ALT-E	Causes physical end of line; display is shifted to beginning of next line, but line is not sent to CPU until Return key is pressed. Useful for entering command lines that are longer than physical line on screen.
ALT-H or Backspace key	Moves cursor back one character position and deletes last character entered.
ALT-J	Terminates input (line feed).
ALT-M or Return key	Terminates input (carriage return).
ALT-R	Redisplays current command line.
ALT-U	Cancels current command line.
ALT-X	Backspaces (deletes) to beginning of current line.
ALT-Z	ASCII end-of-file character; ends input from console.

12.5 FILE SYSTEM

Your computer stores information on diskettes in the form of files; a file is one or a group of related characters. A single file can be any length, up to the data storage capacity of a diskette (6ØØ Kbytes per side); a diskette directory can contain up to 128 entries. All CP/M-86 file-related functions can be used in user-written programs.

12.5.1 FILE SPECIFIERS

A CP/M-86 file is identified by its file specifier, which consists of one to three parts: a file specifier must contain a file name; it may also contain a file extension and/or a drive name (the name of the drive containing the diskette on which the file is recorded). If the drive name is omitted from a file specifier, CP/M-86 assumes that the file is on the diskette in the default drive.

The three elements of a file specifier are arranged in the following sequence, with no intervening spaces:

o Drive name -- one letter followed by a colon.

o File name -- one to eight characters.

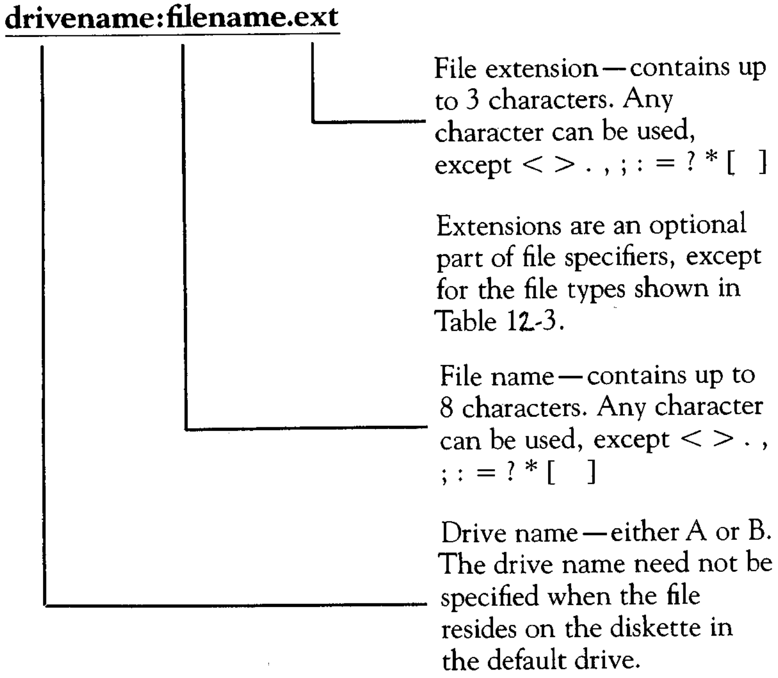
o File extension -- zero to three characters preceded by a period.

Characters in a file specifier can be entered in upper- or lowercase. The following characters cannot be used in a file name or a file extension:

< > . , ; : = ? * [ ]

Examples and command formats in this section present file specifier elements as follows: "filespecifier" represents a file specifier, "drivename" represents a file specifier's drive name, "filename" represents a file name, and "ext" represents a file extension. The three parts of a file specifier are illustrated in Figure 12-1 and are then further described.

Figure 12-1: File Specifier Structure



File extension—contains up to 3 characters. Any character can be used, except < > . , ; : = ? * [ ]

Extensions are an optional part of file specifiers, except for the file types shown in Table 12-3.

File name—contains up to 8 characters. Any character can be used, except < > . , ; : = ? * [ ]

Drive name—either A or B. The drive name need not be specified when the file resides on the diskette in the default drive.

12.5.2 CP/M-86 FILE EXTENSION CONVENTIONS

A file specifier's file extension typically describes the form of the data in the file. Conventions govern some types of files, and some languages and application software packages assign file extensions automatically. For example, MS-BASIC assigns the extension .BAS to programs saved from that language. In addition, CP/M-86 defines the meanings of several file extensions, as presented in Table 12-3.

Files can also be assigned extensions at the operator's discretion. For example, an operator might use the extension .TXT for all text files.

Table 12-3: CP/M-86 File Extension Conventions

FILE EXTENSION	CP/M INTERPRETATION
.A86	8Ø86 assembly language source code
.BAK	Backup file
.BAS	BASIC source code (Microsoft BASIC)
.CMD	Executable command file
.COB	COBOL source code
.DAT	Data file (assumed to be ASCII)
.FOR	FORTRAN source code
.H86	Hex file produced by assembler
.INT	Intermediate compiled code (CBASIC/86, Level II COBOL)
.LST	Listing of compilation or assembly
.OVR	Overlay module
.PRN	Listing of compilation or assembly
.REL	Relocatable object code module
.SUB	Command file executed by SUBMIT command
.SYM	Symbol table of assembly or compilation
.XRF	Cross-reference
.$$$	Temporary, system-generated file

12.5.3 VALID AND INVALID FILE SPECIFIERS

Examples of valid file specifiers follow:

```
PAYROLL
B:1REPORT
PIP.CMD
BJACK.BAS
A:JOE.SMO
```

The following examples present invalid file specifiers; the notations describe why the examples are invalid:

MARCHPAYROL	File name contains more than 8 characters.
B:SALES[2]	File name contains brackets, which are illegal characters.
2:JOE.SMO	Drive name is entered as a number, rather than a letter.

If you attempt to name or save a file with a file name or extension of more than the legal number of characters, CP/M-86 accepts the first eight letters of the file name and the first three letters of the extension as a valid file specifier.

But if you attempt to name or save a file with a file specifier that is invalid in any other way, CP/M-86 responds with the message Invalid Format.

12.5.4 WILD-CARD CHARACTERS

CP/M-86 wild-card characters allow you to use a single command to perform one task on a group of files.

Use of wild-card characters in CP/M-86 is similar to the use of wild cards in card games; the wild-card characters take on the meaning that you assign to them by matching existing file names.

The two CP/M-86 wild-card characters are the question mark (?) and the asterisk (*).

The ? wild-card character means "match any character - or no character -- in this particular location in the file name or file extension." For example, PAY-???.ROL matches each of these file specifiers:

```
PAY-JAN.ROL
PAY-FEB.ROL
PAY-MAR.ROL
PAY-JL.ROL
```

The * wild-card character means "pad with ?s." This enables you to refer to entire families of files. For example, *.BAS refers to all files with the extension BAS. As another example, WS*.* identifies any file whose file name (no matter how long) starts with WS, regardless of extension.

Note that * pads all possible characters that follow it, within the file name or the extension where it appears. This means that CP/M-86 cannot read or match any characters following the * in a file name or extension. For example, *86.CMD matches all files with extension .CMD (????????.CMD), not just files ending in 86 with extension .CMD.

Wild-card characters can be used in combination. For example, TEST?.* refers to all files whose file name is four or five characters long, the first four of which are TEST, regardless of file extension. Note that the file specifier *.* refers to all files on the default or designated drive.

13. CP/M-86 BASIC OPERATING PROCEDURES

This chapter elaborates on some of the operating procedures presented in 12.3 and describes procedures for loading CP/M-86, changing the default drive, and controlling console and printer output via the operating system.

13.1 LOADING CP/M-86

Use the following procedure to load CP/M-86:

1. Be sure both diskette drives are empty.

2. Turn on the system by pressing the switch located at the left rear corner of the processor unit. The LED busy indicator on diskette drive A lights up, and the system displays a power-up display sequence that includes a symbolic request for insertion of the system diskette (e.g., a flashing arrow and a diskette symbol).

3. Insert the system diskette (label up and label-edge last) into drive A and close the drive door. When CP/M-86 is loaded, the system displays a sign-on message (which presents system information, including a keyboard configuration summary) and the CP/M-86 command prompt for the default or logged drive (A>).

13.2. CHANGING THE DEFAULT DRIVE

As described in 12.3, the default drive is the working or currently selected drive. The name of the default drive appears in the CP/M-86 system prompt; for example, the system prompt B> indicates that drive B is the default drive. To change the default drive, respond to the system prompt by entering the letter of the desired drive, a colon

(:), and a carriage return. The following example illustrates changing the default drive to B from A.

```
A>b:(cr)
B>
```

13.3 CHANGING DISKETTES AT THE SYSTEM LEVEL

As a general practice, enter ALT-C whenever you change diskettes at the CP/M-86 system level (when the prompt is >). Entering ALT-C at the > prompt resets or clears the disk system and logs all active drives. Active drives include the default drive, drive A (if it is not the default drive), and any drive specified in CP/M-86 operations since the last loading of the operating system.

CP/M-86 "logs" a drive by building and holding in memory an allocation map of the directory of the diskette in that drive. When you enter ALT-C at the > prompt, the operating system always logs drive A. However, if drive B is the default drive or has been active in CP/M-86 operations, entering ALT-C logs both drives.

If you attempt to write to a newly inserted diskette without first resetting the disk system, CP/M-86 does not perform the write operation and displays the following error message:

```
Bdos Err On X: R/O
```

where X represents the name of the drive containing the diskette. See Section IV -- Solving Problems -- for an explanation of error messages.

13.4 CONTROLLING CONSOLE OUTPUT

When operating at the system level (i.e., when the command prompt is >) and when working with PIP, you can start or stop computer output to either the console or the printer with ALT-P. Starting output to the printer this way causes the printer to "echo" console display. Stopping output to the console temporarily freezes the screen display. Table 13-1 lists the Alternate characters used to control console output.

Table 13-1: Console Output Alternate Characters

ALTERNATE CHARACTER	FUNCTION
^P	Sends all console output to both LST: (for hardcopy) and CON: (for screen display, if CON: is assigned as CRT:). Re-entering ALT-P stops output to LST:. CAUTION: Unless a printer is on-line, entering ALT-P freezes the operating system; the operating system must be reloaded, and all data in memory is lost.
^S	Temporarily stops output to CON:. Console (and printer, if output is being echoed to printer) remains frozen until any key is pressed.

13.5 LOADING CPM-86 OR PROGRAMS FROM DRIVE B

Although CP/M-86 always attempts to load from drive A, the operating system can be loaded from drive B. The option of loading from B is useful if drive A is inoperable for any reason. To load the system from drive B, follow steps 1 and 2 in the procedure in 13.1. Then insert the system diskette in drive B.

After the sign-on message appears, the screen displays this disk operating system error message:

```
Drive = Ø, Track = Ø, Sector = Ø, Error = FA
Bdos Err On A: Bad Sector
```

Enter ALT-C to reset the disk system and to display the system prompt, A>. Then log onto drive B by entering "b:" followed by a Return.

In addition, you can load any CP/M-86 utility or program from a drive other than the default drive by entering the drive name before the command. For example, to load PIP from drive B while working on drive A --

```
A>b:pip(cr)
CP/M-86 PIP VERSION 1.1
*
```

13.6 USING THE HELP COMMAND

The HELP program is an onscreen library of topics explaining CP/M-86 operations, including both resident (built-in) and transient (directory-listed) commands. You can use the HELP command to display explanations and examples for any CP/M-86 commands described in this section.

The format of the HELP command is --

HELP [topic] [subtopic1 subtopic2 ... subtopic8] [P] (cr)

To list the topics available in the HELP library, enter HELP followed by a Return at the system prompt.

To display information about a specific topic, follow HELP with a topic and a Return.

For some CP/M-86 topics the HELP screen display lists available subtopics. To display information about a specific subtopic, follow HELP with both a topic and a subtopic.

The next example illustrates using the HELP command to display the list of topics available in the HELP library.

```
A>help(cr)

HELP UTILITY V1.Ø

At "HELP>" enter topic {,subtopic} ...

EXAMPLE:  HELP>DIR EXAMPLES

TOPICS AVAILABLE:

BOOTCOPY  COMMANDS   TODS       REN
DCOPY     ERA        FILENAME
FORMAT    STAT       SUBMIT

HELP>
```

Entering HELP followed by a Return loads the HELP program, which then displays its prompt, HELP>. After you have loaded the HELP program, you can enter a topic (and subtopics) at the HELP> prompt to display a summary of the topic. You can abbreviate the names of topics and subtopics; one or two letters is enough to identify most topics.

Use the [P] option to prevent the screen display from pausing every 23 lines.

Enter a Return at the HELP> prompt to exit to the CP/M-86 operating system.

The following example uses HELP to display general information about the STAT command and a list of any available subtopics.

```
A>help stat(cr)
```

The next example illustrates a HELP command including a subtopic; the command displays information about STAT command options.

```
A>help stat options(cr)
```

13.7 USING THE TOD (TIME OF DAY) COMMAND

You can examine and set the time of day with the TOD utility. When you start CP/M-86, the date and time are set to the creation date of the BDOS. With the TOD command you can change this initial value to any date and time. You do not have to set time of day for proper operation of CP/M-86.

The command format for TOD is --

```
tod [timespecification]|[P](cr)
```

When you enter the TOD command with no time-specification, the screen displays the current TOD setting. Entering the TOD command with a time-specification changes TOD to the designated setting.

The TOD time-specification follows the format --

```
weekday  month/day/year   hour:minute:second
```

You specify the date with three two-digit values separated by slashes:

- Month value, from 1 to 12 .

- Day value, from 1 to 31, depending on the length of the month.

- Year value, the last two digits of the year, from ØØ to 99.

You specify time as on a twenty-four hour clock, with hour values ØØ to 11 for the morning, and 12 to 23 for the afternoon. Separate the three time values with colons.

The following example illustrates a screen display in response to the TOD command.

```
A>tod(cr)
12/Ø6/81        Ø9:15:37
A>
```

To set the date and time, use the command form --

```
tod timespecification(cr)
```

The next example sets the current date and time to the last second of 1982.

```
A>tod 12/31/82 23:59:59(cr)
```

To let you set the time accurately, the TOD utility displays the message:

```
Press any key to set time
```

When the time that you specify in the command occurs, press any key. TOD begins timing from that instant.

Use the P option to continuously print the date and time on the screen. You can stop the continuous display by pressing any key.

14. DEVICE AND MEDIA MANAGEMENT

CP/M-86 includes these four programs for managing diskette files:

- FORMAT formats diskettes for system use.
- DCOPY copies a diskette's contents onto another diskette.
- BOOTCOPY copies the operating system tracks from one diskette to another.
- STAT reports statistics and controls certain characteristics of diskettes and files.

14.1 DISKETTE FORMATTING — FORMAT

The FORMAT program prepares diskettes to receive data. In the process, FORMAT automatically erases any previous files on the diskette. New diskettes must be formatted before they can be used by the system.

14.1.1 USING FORMAT

1. Insert the system diskette in drive A and enter --

   format(cr)

   Your computer loads FORMAT and displays the utility's one-line sign-on banner near the top of the screen.

2. Once FORMAT is loaded, you can remove the system diskette. Insert the diskette to be formatted in either drive.

3. At the bottom of the screen, FORMAT asks which drive contains the diskette to be formatted, as follows:

   Format drive? (A or B; press return key to end.)

   Enter the letter name of the correct drive.

4. The system responds by displaying --

   Format drive X. Press space bar when ready.

   where X is the drive name you entered in step 3.

   Check that the diskette to be formatted is in the correct drive (to avoid possible data loss) and press the Space bar.

5. The formatting procedure takes approximately one minute for a single-sided diskette. At the bottom of the screen, FORMAT displays the number of each track as it is formatted.

   Near the top of the screen, FORMAT displays the message --

   Format drive X

   where X is the drive name you entered in step 3.

6. When formatting is complete, near the top of the screen, FORMAT displays --

   Format drive X complete.

7. FORMAT prompts you to repeat the process. To format another diskette, repeat steps 2 through 5. To end the FORMAT program, insert the system diskette in drive A, and enter ALT-C or press the Return key.

The following example illustrates a sample session using FORMAT to format one diskette. The example shows FORMAT prompts and messages consecutively, as they appear on the screen, but does not indicate where on the screen the prompts and messages are displayed.

```
A>format(cr)

Diskette FORMAT Utility - Version 2.9

Format drive? (A or B; press return key to
end.)b

Format drive B.  Press space bar when
ready.(space)

Format drive B

Format drive B complete.

Format drive? (A or B; press return key to
end.)(cr)

A>
```

14.1.2 USING FORMAT SWITCHES

You can expand the FORMAT command by including a drive name and optional "switches" that modify FORMAT program operation. These additions must appear in the command line with the FORMAT command stem; a switch value or group of switches must be preceded by a $ (dollar sign). There are four switches:

C -- Display the count of the tracks formatted and the number of soft errors encountered.

E -- Display the locations of soft errors encountered.

Z -- Display disk zone information (size of tracks and gaps).

D -- Format a double-sided diskette.

Tracks are circular sections of a diskette; your computer's diskettes have 8Ø tracks on a side. These tracks are grouped into eight zones; the drive motor runs at a different speed on each zone. Soft errors are hard-to-read sections or worn or flawed spots on the diskette that may make it unreliable for use. The $C switch (for counting tracks and soft errors) allows you to gauge the reliability of a diskette for recording data. If 15 to 2Ø or more errors appear, repeat the FORMAT process a few times. If this number of errors persists, discard the diskette and try another.

One to four switches may be included in the FORMAT command line, as follows:

```
format [drivename:] [$c|e|z|d](cr)
```

The next example presents the command to format a disk in drive B and then to display (1) the number of tracks copied and (2) the number and the locations of soft errors encountered in the process. (See Chapter 22 for explanation of FORMAT error messages.)

```
A>format b: $ce(cr)

Soft format error: D=1, T=Ø1, S=12, E=4C
Soft format error: D=1, T=Ø1, S=FF, E=41
Soft format error: D=1, T=Ø1, S=Ø6, E=4A

Format complete.
8Ø tracks formatted: 3 soft errors.

A>
```

14.2 DISKETTE BACKUP — DCOPY

The DCOPY program copies the contents, including the system tracks, of one diskette onto another diskette, creating a literal twin of the source or copy-from diskette. In the process, DCOPY formats the destination or copy-to diskette for CP/M-86 (eliminating the need to run the FORMAT program separately).

There are two methods -- long and short -- for using DCOPY. With the long method, the program prompts you for the names of the copy-from and copy-to drives. With the short method, you enter the program name and the copy-from and copy-to drive names in a single command line, without prompts from DCOPY.

You can use the long DCOPY method (described in 14.2.1) to copy the contents of one or more diskettes in either drive and to make more than one copy. You can use the short method (described in 14.2.2) to make a single copy of the diskette in the default drive; the copy-from diskette must contain DCOPY. With this method, the system exits the program immediately after the copy is completed.

Several of the optional switches described for the FORMAT program can be included in the DCOPY command (see 14.1.2 and 14.2.2).

14.2.1 USING DCOPY WITH INTERACTIVE SYSTEM PROMPTS

Use the following procedure to copy a diskette from either drive and to make multiple copies.

1. Insert the system diskette in drive A, and enter the following command:

   dcopy(cr)

   Your computer loads DCOPY and displays the utility's one-line sign-on banner at the top of the screen.

2. Remove the system diskette from drive A if it is not the copy-from diskette.

3. At the bottom of the screen, DCOPY asks for the name of the drive containing the diskette to be copied:

   Copy from drive? (A or B; press return key to end.)

   Insert the copy-from diskette in either drive and insert the copy-to diskette in the other drive. Answer the DCOPY query with the correct copy-from drive name. DCOPY assumes that the remaining drive is the copy-to drive.

4. DCOPY repeats the particulars of the copy command and asks for confirmation:

   Copy from drive f to drive t. Press space bar when ready.

   where f is the name of the copy-from drive, and t is the name of the copy-to drive. Press the Space bar to start the DCOPY process. Press the Return key to cancel the DCOPY command.

5. During the copy process, at the bottom of the screen, DCOPY displays the number of each track as it is copied.

Near the top of the screen, DCOPY displays the message --

Copy from drive f to drive t

When the copy is complete, the system prompts with:

Copy from drive f to drive t complete.
Copy from drive? (A or B; press return key to end.)

6. To make another copy, follow steps 3 and 4 above.

   To exit DCOPY (return to the operating system), insert the system diskette in drive A and enter ALT-C or press the Return key.

The following example illustrates a sample session using DCOPY to copy one diskette. The example shows DCOPY prompts and messages consecutively, as they appear on the screen, but does not indicate where on the screen the prompts and messages are displayed.

```
A>dcopy(cr)

Diskette COPY Utility - Version 2.6

Copy from drive? (A or B; press return key to
end.)a

Copy from drive A to drive B.  Press space
bar when ready.(space)

Copy from drive A to drive B

Copy from drive A to drive B complete.
Copy from drive? (A or B; press return key to
end.) (cr)

A>
```

14.2.2 USING DCOPY WITHOUT SYSTEM PROMPTS

Use the following procedure for the short, one-command DCOPY method.

1. To copy the system diskette or any diskette containing DCOPY, insert the copy-from diskette in the default drive, and insert the copy-to diskette in the other drive. Then enter the following command:

   ```
   dcopy f: to t:(cr)
   ```

   where f represents the copy-from drive name, and t represents the copy-to drive name.

2. At the bottom of the screen, DCOPY displays the number of each track as it is copied. When DCOPY has completed the copy process, the system displays the following message:

   ```
   Copy from drive f to drive t complete.
   ```

NOTE: DCOPY's short command form can include the switches D, C, E, and Z, as for the FORMAT program (see 14.1.2). For example, to display the count of tracks copied and of soft errors encountered, and to copy the contents of a diskette in drive A to a diskette in drive B, enter --

```
A>dcopy a: to b: $c(cr)
```

14.3 OPERATING-SYSTEM COPY - BOOTCOPY

The CP/M-86 BOOTCOPY program creates a system diskette by copying the system tracks from a system diskette onto another diskette. The system tracks contain the portion of the operating system that "boots" or loads into memory when you insert the system diskette at power-up or when you press the Reset button to re-load the operating system. The system tracks also contain the resident CP/M-86 commands, such as DIR and TYPE, that do not appear on the diskette directory.

For BOOTCOPY to run, the destination diskette must have been formatted with the CP/M-86 FORMAT program. If the destination diskette already contains an operating system, BOOTCOPY writes over it. The BOOTCOPY process does not affect any files on either diskette's directory.

Like DCOPY, the BOOTCOPY program has both a short and a long command form (see 14.2). The following procedure and example depict the long form of BOOTCOPY (with interactive queries from the program). The short, one-command version is described at the end of this discussion.

To copy the operating system tracks from one diskette to another, use the following procedure:

1. Load the operating system (if it is not already loaded) and enter the following command:

   bootcopy(cr)

   After loading, the BOOTCOPY program identifies itself with a sign-on banner and then asks which drive contains the source or copy-from diskette:

   Source drive (A/B) ?

2. Enter the source drive name. The program identifies the selected drive and requests confirmation:

   Source disk in drive X and press <space>.

   where X is the name of the drive entered in step 1.

   Make sure that drive X contains the diskette whose system tracks you want to copy.

3. Press the Space bar to begin the BOOTCOPY process. BOOTCOPY reads the operating-system sectors into memory and then asks which drive contains the destination or copy-to diskette:

   Destination drive (A/B) ?

4. Place the destination diskette in either drive and enter the drive name. BOOTCOPY displays the following request for confirmation:

   Destination disk in drive X and press <space>.

   where X is the name of the drive entered in step 3. Press the Space bar to complete the BOOTCOPY process.

5. When the copy is complete, BOOTCOPY displays a confirming message and again prompts for a destination drive:

   Bootcopy complete.
   Destination drive (A/B) ?

   If you want to BOOTCOPY the same system tracks to another diskette, insert another destination diskette and follow step 4 above.

To exit the program, press the Return key.

To BOOTCOPY from a different source diskette, exit the program and then enter a new BOOTCOPY command.

The following example illustrates a sample session of BOOTCOPY.

```
A>bootcopy(cr)

BOOTCOPY  Version 1.6

Source drive (A/B) ?a
Source disk in drive A and press
<space>.(space)

Reading system sectors...

Destination drive (A/B) ?b
Destination disk in drive B and press
<space>.(space)

Writing system sectors...

Bootcopy complete.

Destination drive (A/B) ?(cr)

A>
```

The command format for the short, one-command version of BOOTCOPY follows:

```
bootcopy [drivename: to drivename:](cr)
```

With the short command form, BOOTCOPY exits to CP/M-86 after completing one copy. To BOOTCOPY again, enter another BOOTCOPY command.

14.4 DEVICE AND FILE STATUS — STAT

Use the STAT program to display information about diskettes and files, to specify whether file names are to be displayed in the directory, to specify whether diskette contents can be user-modified, and to change physical-to-logical device assignments.

Parameters specified in the STAT command determine what information about the system, the diskettes, or the files is displayed. Refer to Table 14-1 for STAT commands, parameters, and formats.

Table 14-1: STAT Command Options

COMMAND/PARAMETER	BRIEF DESCRIPTION
stat(cr)	Displays amount of available storage space on diskette in default or active drive(s).
stat dev:(cr)	Displays physical-to-logical device assignments.
stat drivename:(cr)	Displays amount of available storage space in specified drive.
stat [drivename:] dsk:(cr)	Displays diskette characteristics for active drive(s) or specified drive.

Table 14-1: (Continued)

COMMAND/PARAMETER	BRIEF DESCRIPTION
stat drivename:=ro(cr)	Temporarily sets a drive to read-only; enter ALT-C to restore RW status.
stat filespecifier(cr)	Displays file characteristics and read-write status.
stat filespecifier [size] (cr)	Displays file size and read-write status.
stat filespecifier {[ro]|[rw]}(cr)	Sets file access attribute to read-only (RO) or read-write (RW).
stat filespecifier {[sys]|[dir]}(cr)	Makes file a system file or a directory file.
stat logicaldevice=physicaldevice(cr)	Changes physical-to-logical device assignments
stat val:(cr)	Lists available options for STAT command parameters.

14.4.1 AVAILABLE SPACE ON ALL DRIVES

When entered with no parameters, the STAT command displays the amount of available disk space and the read/write status of all active diskette drives.

Active drives, as defined in 13.3, include the default drive and any drive previously specified in CP/M-86 operations. As an illustration, once the DIR B: command has displayed the directory of drive B, that drive is an active drive even if it is not the default drive.

To display file information concerning the diskettes in drives A and B when both drives are active, enter the following command:

```
stat(cr)
```

The following example demonstrates using STAT when drive A contains a diskette with 12ØK bytes of available storage and drive B contains a read-only diskette with 215K bytes of available storage.

```
A>stat(cr)
A: RW, Free Space: 12Øk
B: RO, Free Space: 215k

A>
```

14.4.2 AVAILABLE SPACE ON A SPECIFIED DRIVE

The STAT command followed by a drive name displays space information for the diskette in the specified drive.

The following example illustrates using STAT to display the statistics for a diskette in drive B that contains 12ØK bytes of available memory.

```
A>stat b:(cr)
B: RW, Free Space:    12Øk

A>
```

14.4.3 FILE INFORMATION

When followed by a file specifier, the STAT command displays information about the specified file(s). Wild-card characters (? and *) can be used as described in 12.5.4.

The CP/M-86 STAT display uses the following column heads:

Recs – The number of 128-byte records in the file

Bytes – The number of 8-bit bytes (where K means 1Ø24) allocated to the file

FCBs – The number of File Control Blocks for the file

Attributes – The accessing attributes of the file, which indicate whether the file is a Directory or a System file, and whether the file is read-only or read-write.

The next example demonstrates using STAT to display file information for all files on the diskette in the logged drive that have names beginning with S and extensions of .CMD.

```
A>stat s*.cmd

Drive A:                                 User   :  Ø
     Recs   Bytes  FCBs Attributes        Name
     388    5Øk     4 Dir RW           A:SETBPS  .CMD
      73    1Øk     1 Dir RW           A:STAT    .CMD
      31     4k     1 Dir RW           A:SUBMIT  .CMD
     ---------------------------------------------------
Total:      64k     6
A:  RW, Free Space:     216k

A>
```

You can add an additional column, Size, to the STAT display by including the [SIZE] parameter. With random (sparse) files the final record number measured with [SIZE] may be much larger than the actual number of records allocated. The next example illustrates using STAT to display size information for a random file.

```
A>stat test.dat [size](cr)

Drive A:                                     User   :  Ø
    Size   Recs   Bytes  FCBs Attributes      Name
   1ØØØØ    17     4k     2 Dir RW         A:TEST .DAT
---------------------------------------------------------
Total:                4k     2
A: RW, Free Space:      154k
A>
```

14.4.4 SETTING A FILE OR DISKETTE TO READ-ONLY OR READ-WRITE STATUS

As indicated in 14.4.3, if you set the accessing attribute of a diskette or file to read-only, the data in the file or diskette cannot be written over or erased (except with FORMAT or DCOPY). If a diskette or file is set to read-write status, you can modify or erase the data.

The command format to change a file's accessing attribute follows:

```
stat filespecifier {[ro]|[rw]}(cr)
```

Note that the brackets around RO and RW are literal characters that you enter in the command.

The following example sets the file JOE.TXT on drive B to read-only.

```
A>stat b:joe.txt [ro](cr)

B:JOE     .TXT set to Read Only (RO)
A>
```

You cannot write to a file that has been write-protected with the STAT [RO] commnd until you reset the file's accessing attribute with a STAT [RW] command. The following example illustrates setting the file JOE.TXT to read-write status.

```
A>stat b:joe.txt [rw](cr)

B:JOE     .TXT set to Read Write (RW)
A>
```

The command format to set a drive to temporary read/only status follows:

```
stat drivename:=ro (cr)
```

where drivename is the name of the drive containing the diskette you want to temporarily set to RO status.

The next example demonstrates temporarily setting drive B to read-only status.

```
A>stat b:=ro(cr)
```

To restore a diskette to read-write status, enter ALT-C at the system prompt.

You can ensure permanent write-protection for a diskette by using a write-protect tab. Covering the write-enable notch on the diskette jacket with an adhesive write-protect tab prevents writing to the disk until the tab is removed.

14.4.5 EXCLUDING FILE NAMES FROM THE DICTIONARY

You can use STAT to exclude file names from the diskette directory display. The excluded files, known as Sys (system) files, are still available for use although they are not listed by the DIR command. To see the current directory of system files, enter the DIRS command (see 15.1).

Use the STAT [SYS] command to set executable files, such as .CMD (command) files, to Sys status; these files can then be used in different USER areas (see 15.7). Also, in the STAT display (illustrated in 14.4.3) the Attribute of system files is listed as Sys.

Use the following command format to exclude a file name from the diskette directory:

```
     stat filespecifier [sys](cr)
```

To reinstate a file name in the diskette directory, use the following command format:

```
     stat filespecifier [dir](cr)
```

14.4.6 DISPLAYING STAT COMMAND PARAMETERS

Enter the following command to display all the parameters that can be manipulated with STAT:

```
stat val:(cr)
```

The system responds with a listing of the CP/M-86 STAT parameters, as follows:

```
A>stat val:(cr)
STAT 2.2

Read Only Disk:  d:=RO
Set Attribute: d:filename.typ [RO] [RW] [SYS] [DIR]
Disk Status :  DSK: d:DSK
User Status :  USR: d:USR:
Iobyte Assign :

CON: = TTY: CRT: BAT: UC1:
AXI: = TTY: PTR: UR1: UR2:
AXO: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:

A>
```

14.4.7 DISPLAYING DRIVE CHARACTERISTICS

To display disk drive characteristics, enter the STAT command followed by a disk drive name and the special term, DSK:. The system responds with a listing of the sizes of the tracks, blocks, and sectors on the diskette, as well as the capacity of the diskette and its directory.

The following example illustrates using STAT to display the disk drive characteristics of a diskette in drive B.

```
A>stat b:dsk:(cr)

      B: Drive Characteristics
   488Ø: 128 Byte Record Capacity
    61Ø: Kilobyte Drive Capacity
    128: 32 Byte Directory Entries
    128: Checked Directory Entries
    128: Records/Directory Entry
     16: Records/Block
     64: Records/Track
      Ø: Reserved Tracks
A>
```

14.4.8 PHYSICAL-TO-LOGICAL DEVICE ASSIGNMENTS

The STAT command assigns a physical device to a logical device name, within allowable physical-to-logical device relationships, as listed in Table 14-2. Four logical devices appear in the first table column. For each logical device, you can assign any of the four physical device names listed beside it in the second column. Physical devices and logical device names are described in 12.2.

Table 14-2: Physical-to-Logical Device Assignments

LOGICAL DEVICE	ASSIGNABLE PHYSICAL DEVICES			
CON:	TTY:	CRT:	BAT:	UC1:
LST:	TTY:	CRT:	LPT:	UC1:
AXI:	TTY:	PTR:	UR1:	UR2:
AXO:	TTY:	PTP:	UP1:	UP2:

One STAT command can assign all four logical devices, provided they are separated by single commas. The command format for assigning physical to logical devices follows:

```
stat logicaldevice=physicaldevice [,...](cr)
```

The next example demonstrates changing the auxiliary output device to TTY:, changing the list device to LPT:, and changing the auxiliary input device to TTY:.

```
A>stat axo:=tty:,lst:=lpt:,axi:=tty:(cr)
```

Use the STAT command with the DEV: parameter to check the physical/logical device assignments. The next example demonstrates displaying the four logical device assignments.

```
A>stat dev:(cr)
CON: is CRT:
AXI: is TTY:
AXO: is TTY:
LST: is UL1:

A>
```

15. FILE SYSTEM MANAGEMENT

CP/M-86 provides six system-level resident or built-in commands and two utility programs that assist in managing files:

- The DIR command displays the file names and extensions for the Directory files stored on the logged or specified diskette(s).
- The DIRS command displays file names and extensions of System files.
- The TYPE command displays the information stored in ASCII files.
- The ERA command deletes files from a diskette.
- The REN command renames an existing file.
- The PIP program (the Peripheral Interchange Program) moves files (or file parts) from one storage device or media to another.
- The SUBMIT program allows you to string several commands together in one file.
- The USER command displays and changes the current user number.

15.1 THE DIR AND DIRS (DIRECTORY) COMMANDS

The DIR and DIRS commands display the directory listing of file specifiers for files and programs on the diskette in the default or designated drive. Both commands accept wild-card characters in the file specification, if any is included.

The DIR command lists the names of files in the current user number that have the Directory (Dir) attribute (displayed by the STAT command in 14.4.3).

The DIRS command displays the names of files in the current user number that have the System (Sys) attribute. Although you can access Sys files stored in user Ø from any other user number on the same drive, DIRS displays user Ø Sys files only if the current user number is Ø.

The format of the DIR and DIRS commands is --

dir|dirs [drivename:|filespecifier](cr)

If the DIR (or DIRS) command is entered without a drive name or file specifier, the system displays the directory of Dir (or Sys) files for the diskette in the logged drive.

If a drive name is specified, the system displays the directory for the diskette in that drive. If a file specifier is given, the system lists the file or files (wild-card characters can be used in the file specifier) that match it.

If the diskette directory is empty or if DIR (or DIRS) cannot find a file specifier that matches the given file specifier, the system responds with a message such as NO FILE.

If System (Sys) files reside on the specified drive, DIR displays the message --

SYSTEM FILE(S) EXIST

If non-system (Dir) files reside on the specified drive, DIRS displays the message --

NON-SYSTEM FILE(S) EXIST

You cannot use a wild-card character in a drive specifier.

To display the file directory for the diskette in the default drive, enter the following:

```
     dir(cr)
```

To list the names of all System (Sys) files in the current user number on drive A, enter —

```
     dirs a:(cr)
```

The system responds as shown in the following hypothetical example:

```
A>dir(cr)
A: PIP      CMD : STAT  CMD : ASM86 TXT : SUBMIT CMD
A: BOOTCOPY CMD : DCOPY CMD : ED    CMD : FORMAT CMD

A>
```

Three DIR command line examples follow. To display all files on the diskette in drive B (regardless of which drive is currently logged) —

```
     A>dir b:(cr)
```

The following example illustrates using DIRS to display the names of all Sys files with a file extension of .CMD in the current user number on drive A.

```
     A>dirs *.cmd(cr)
```

To verify the existence of a Directory file named ACCRED.INT on the diskette in drive B —

```
     A>dir b:accred.int(cr)
```

15.2 THE TYPE (DISPLAY) COMMAND

The TYPE resident command displays the contents of an ASCII text file. Use ALT-S to stop and start the display. You must give a specific file name; wild-card characters cannot be used with TYPE.

The format of the command follows:

```
type filespecifier(cr)
```

Two command line examples follow. To display a file named SUMMARY on the diskette in the default drive —

```
A>type summary(cr)
```

To display the file named MEMO.BAK on the diskette in drive B —

```
A>type b:memo.bak(cr)
```

15.3 THE ERA (DELETE) COMMAND

The ERA resident command deletes the directory entries of one or more files. Unlike the FORMAT program, which erases the entire diskette, ERA deletes only the directory entry, thereby freeing previously allocated file space for re-use.

Wild-card characters can be used (with caution) to delete groups of files with a single ERA command.

The format of this command follows —

```
era filespecifier(cr)
```

If CP/M-86 does not find a file with the given file specifier, the system displays the message NO FILE.

The following example shows erasing a file named

ACCOUNTS.BAS on a diskette in the default drive.

```
A>era accounts.bas(cr)
```

You can erase all files on a diskette with the *.* wild-card characters, as in the next example.

```
A>era *.*(cr)
```

The system responds to the preceding command (to erase all files on a diskette) with the message —

```
ALL (Y/N) ?
```

Enter Y to erase all files on the logged or specified drive(s). Any other response cancels the command.

15.4 THE REN (RENAME) COMMAND

The REN resident command changes the directory entry (the file specifier) a file. The format of the command follows:

```
ren newfilespecifier=oldfilespecifier(cr)
```

Wild-card characters cannot be used with REN. If the new file specifier named in the REN command already exists, the system responds with the message FILE EXISTS and cancels the command.

Each REN operation takes place on one drive only — either the default drive or the drive named in the new file specifier. If the REN command line names two different drives, the system responds with a non-recognition query and cancels the command.

For example, to rename ACCOUNTS.BAS as BOOKS.BAS on the default drive —

```
A>ren books.bas=accounts.bas(cr)
```

To rename PIP.CMD to COPY.CMD on drive B —

```
A>ren b:copy.cmd=pip.cmd(cr)
```

15.5 PIP — PERIPHERAL INTERCHANGE PROGRAM

You can use PIP and its command parameters for many file-handling functions. PIP copies, prints, displays, and combines diskette files. PIP can read one or more source files and copy them individually (with the same or new file specifiers) onto another diskette. PIP can concatenate (combine) several files into a single file on the source diskette or on another diskette. It can also copy a portion of a file to another file. Further, PIP can output a file to a peripheral device such as a CRT or printer. Optional PIP command parameters, described in Table 15-1 and in 15.5.3, can modify data as it is copied.

15.5.1 LOADING PIP

You can enter PIP commands in either of two ways. The first method is to enter the the PIP command stem, followed by a Return (without any file specifiers or parameters). Once loaded in this manner, the PIP program displays its prompt, an asterisk (*), to indicate that it is ready to accept specifiers and parameters. The PIP program remains loaded until ALT-C or a Return — entered as a complete command line -- returns the system to CP/M-86.

This method, as illustrated in the next example, allows you to perform a number of PIP operations without re-entering the command stem and reloading PIP for each one.

```
A>pip(cr)
CP/M-86 PIP VERSION 1.1
*
```

The second method of loading the PIP program is to use one command line to enter the PIP command stem, file specifiers, and parameters for the operation you are requesting. Pressing the Return key loads PIP, enters file specifiers and parameters, and executes the entire command. After execution, control of the system returns to CP/M-86 (which displays a >). To run PIP again, you must re-enter the PIP command stem plus the command particulars for the next operation.

Except for 15.5.2, the examples in this section show the second (one-line) method of entering PIP commands.

15.5.2 COPYING FILES WITH PIP

The general format for PIP commands that copy files follows. Note that you should enter PIP parameters inside square brackets. In the PIP command format, the first pair of brackets around parameterlist are a typographical device that indicates optional elements (see 11.2, "Section Conventions"). The second pair of brackets are literal characters that you enter in the command line. The use of PIP parametes is discussed in 15.5.3.

```
pip destination=source[,source2..]
    [[parameterlist]](cr)
```

The definitions of the three general forms in the PIP command format follow:

Destination -- the name of the file or device that will receive the data; destination can include a disk drive or a logical device name followed by a colon. If the destination is a file that already exists, PIP overwrites the file; if the destination is a file that does not exist, PIP creates a file with that name.

Source -- the name(s) of the file(s) or device(s) that will be copied onto the destination; source can include a disk drive or device name followed by a colon.

Parameterlist -- one or more command parameters (described in Table 15-1) enclosed in brackets, immediately following the name(s) of the affected file(s) or device(s).

The following examples illustrate using PIP to make a copy of file X, calling the new file Y, all on the logged drive. The first example also illustrates using the one-line method for loading and executing PIP. The second example illustrates the multiline method.

```
A>pip y=x(cr)
A>
```

```
A>pip(cr)
CP/M-86 PIP VERSION 1.1
*y=x(cr)
*(cr)
A>
```

You can abbreviate PIP commands under some circumstances. Only one file specifier requires a file name when a file is copied from one drive or device to another with no change in the file name itself; CP/M-86 assumes the file name to be the same for both destination and source.

The next three examples demonstrate alternative abbreviated command lines that can be entered to copy file JOE.TXT from the logged drive (A) to drive B, without changing the file name. Note that CP/M-86 always assumes the default drive in any file specifier that contains no specific drive name.

```
A>pip b:=joe.txt(cr)

A>pip b:joe.txt=a:(cr)

A>pip b:=a:joe.txt(cr)
```

The wild-card character * can be used to manipulate groups of files. For example, to copy all files with an extension of .CMD from drive A (as the default drive) to drive B, without changing the file names —

```
A>pip b:=*.cmd(cr)
```

The next example demonstrates file concatenation (combination) with PIP. Files X, Y, and Z on drive A are concatenated into one file named Q on drive B. Note that the drive name is included in each file specifier.

```
A>pip b:q=a:x,a:y,a:z(cr)
```

15.5.3 USING PIP PARAMETERS

You can modify copy operations by including command parameters in brackets in the PIP command. (15.5.1 describes the use of brackets in PIP commands.) Table 15-1 lists the PIP command parameters in CP/M-86; examples illustrating these parameters follow the table.

Table 15-1: PIP Command Parameters

PARAMETER	EFFECT
Dn	Deletes characters extending past column n. Use this parameter to truncate long lines sent to printer or display.
E	Echoes or outputs to the console the characters in the file affected by the PIP operation.
F	Filters or removes the existing form feeds from the file. Use the P parameter in the same PIP parameter list to insert new form feeds.
Gn	Go to user number n.
H	Transfers hex data. Checks all data for proper hex file format. Removes non-essential characters between hex records. System then prompts for corrective action if errors occur.
I	Ignores :ØØ records in transfer of hex format file. (I parameter automatically sets H parameter.)
L	Translates upper case letters to lower case.
N	Numbers each line, starting at 1 and incrementing by 1. Suppresses leading zeros; line numbers are followed by colons. If N2 is specified, includes leading zeros and inserts tabs following the line numbers. Expands tabs if T parameter is set.

Table 15-1 (Continued)

PARAMETER	EFFECT
O	Transfers object files (non-ASCII); normal CP/M-86 end-of-file terminator is ignored.
Pn	Includes one form feed (page eject) every n lines (with initial form feed). If n equals 1 or is omitted, form feeds occur every 6Ø lines. If F parameter is used, deletes existing form feeds before new ones are inserted.
Qs^Z	Stops copying from source device or file when string s is encountered. If no Q parameter is entered, copying stops at end-of-file. Use S parameter to specify the string from which to start copying.
R	Reads system files (files with directory display suppressed by STAT [SYS] command).
Ss^Z	Starts copying when string s is encountered. Use S and Q parameters (start and quit strings) to select a portion of a file for copying. Copying will start at the file beginning if no S parameter is entered. PIP translates start and quit strings into upper case if the user enters the string on the same line as the PIP command itself. Strings are not translated if the string is entered in response to the PIP prompt (*).

Table 15-1 (Continued)

PARAMETER	EFFECT
Tn	Expands tabs (ALT-I characters) to every nth column.
U	Translates lower case letters to upper case.
V	Verifies that data has been copied correctly by rereading after write operations (destination must be file).
W	Writes over RO files without console interrupts.
Z	Forces parity (high order) bit of each ASCII character to zero.

Four examples illustrating the use of PIP parameters follow.

To make a copy of X.TXT, renaming it Y.TXT (when both files reside on drive B) and to instruct PIP to echo the file to the console, translate lower case letters to upper case, and verify the results —

```
A>pip b:y.txt=b:x.txt[euv](cr)
```

To copy a portion of file Y.TXT on drive B onto file X.TXT on drive A, starting at the words "Once upon" and continuing to the end of the file —

```
A>pip a:x.txt=b:y.txt[sOnce upon^Z](cr)
```

You can use PIP to create and format disk files so that files entered with a text editor or word processing program can be printed within CP/M-86. For example, to output or copy the file X.TXT from drive B to the LST: device, expand tabs every 8 columns, and start a new page every 60 lines (standard text-entry format translated into CP/M-86 PIP parameters) —

```
A>pip lst:=b:x.txt[t8p60](cr)
```

To copy the file X.TXT on drive B to the diskette in drive A (keeping the same file name), delete the embedded form feeds, change the page length to 52 lines, and number each line —

```
A>pip a:=b:x.txt[nfp52](cr)
```

15.5.4 OUTPUTTING FILES TO LOGICAL DEVICES WITH PIP

PIP can "copy" or output files to logical devices as well as to storage media. Table 15-2 lists and describes the logical devices and the special terms you can use with PIP; examples follow Table 15-2 to illustrate the use of these terms.

When a logical device is specified as the source, PIP reads output from that device until encountering an end-of-file marker. When copying a keyboard entry to a logical device, enter ALT-Z to indicate end-of-file.

When a logical device is specified as the destination, PIP copies or outputs the specified file to that device. For example, specifying the logical device CON: as the destination in a PIP command will display the file contents on the console screen.

The specification of logical devices in PIP commands is subject to physical constraints. When a logical device name appears in either the destination or source fields of a PIP command, the physical device assigned to that logical device (see 14.4.8) must be able to receive or send data as specified. For example, the logical device LST: is an output device and cannot be a source of data.

Table 15-2: Logical Devices/Special Terms Used with PIP

DEVICE	FUNCTION
CON:	Sends the specified data to the console device.
LST:	Sends the specified data to the list device.
AXI:	Reads the specified data from the auxiliary input device.
AXO:	Sends the specified data to the auxiliary output device.
PRN:	Similar to LST:, except that tabs are expanded at every eighth character position, lines are numbered, and form feeds are inserted at the beginning of the file and at every 6Øth line. PRN: is equivalent to LST:[t8np6Ø].
EOF:	Sends CP/M-86 end-of-file marker (ALT-Z) to destination device.
NUL:	Sends 4Ø nulls (ASCII Øs) to destination device (this can be issued at the end of punched output).

To concatenate three .A86 files and output them to the CON: device, followed by an end-of-file marker --

```
A>pip con:=x.a86,y.a86,z.a86,eof:(cr)
```

15.5.5 BACKING UP FILES WITH PIP

You can use PIP to copy a working file onto a backup diskette.

If either diskette contains PIP, insert the PIP diskette (containing PIP.CMD) in the default drive and the second diskette in the other drive. Then reset the disk system (type ALT-C at the system prompt), and enter the appropriate PIP command (see 15.5.2).

If neither the working nor the backup diskette contains PIP, the backup procedure involves loading PIP from a third diskette and changing diskettes after the PIP prompt appears.

Note that if you attempt to write (or PIP) onto a newly inserted diskette without first clearing the disk system (entering ALT-C at the system prompt), the screen displays the following error message:

```
Bdos Err on X:R/O
```

where X is the name of the drive containing the newly inserted diskette.

To avoid this error condition, clear the disk system with the backup diskette in the destination drive before you enter the PIP command.

Use the following procedure to copy a file from one diskette to another when neither diskette contains PIP.

1. Insert the destination or backup diskette in drive B.

2. Insert a diskette containing PIP.CMD in drive A. Enter ALT-C at the system prompt to clear the disk system in preparation for writing to the diskette in drive B.

3. At the system prompt, enter PIP followed by a Return. After the PIP prompt appears, replace the diskette in drive A with the copy-from or source diskette.

4. Complete the PIP command by supplying the destination and source file specifiers (see 15.5.2).

The following example demonstrates how to copy the file TEXT.DOC onto a backup diskette in drive B when neither the working nor the backup diskette contains PIP.

SCREEN DISPLAY	PROCEDURE
	(Insert PIP diskette and destination diskette in drives A and B, respectively.)
A>^C A>pip(cr) CP/M-86 PIP VERSION 1.1	
	(Remove PP diskette and insert source diskette in drive A.)
*b:=a:text.doc(cr) *(cr) A>	

15.6 THE SUBMIT UTILITY

SUBMIT allows you to execute the commands stored in a file. The file must be named with the .SUB extension. The file name in the SUBMIT command names the .SUB file (you need not type the .SUB extension). You use a text editor to create the file listing the commands to be executed.

The SUBMIT command format is:

SUBMIT [drivename:]filename [parameterlist](cr)

There are several drive requirements for a .SUB file to be executed:

- The file SUBMIT.CMD must be on the default drive for SUBMIT to run.

- The commands listed in the .SUB file must be on drive A, unless a drivename is included with the command.

- The diskette in drive A must be R/W (read/write).

- The .SUB file may be on either drive; type the optional drive name if the file is not on the default drive.

The commands in the .SUB file can contain substitution parameter numbers. The numbers are identified with a dollar sign ($1, $2, etc.). When the .SUB file is executed, the parameters listed with the SUBMIT command are substituted in order for the parameter numbers. If the .SUB file contains no substitution parameters, you cannot vary the contents of the submitted file.

Typing another SUBMIT command as the last command in a .SUB file creates chained command files.

For example, suppose the file ASMBL.SUB contains the following CP/M-86 commands:

```
ASM86 $1
DIR $1.*
ERA *.BAK
PIP $2:=$1.PRN
ERA $1.PRN
```

To execute the commands listed in file ASMBL.SUB and to substitute REPORT1 for $1 and PRN for $2 in the .SUB file, type the command:

```
SUBMIT ASMBL REPORT1 PRN(cr)
```

SUBMIT reads the .SUB file, substitutes REPORT1 for all occurrences of $1 and PRN for $2, and creates a file named $$$.SUB that contains the following commands:

```
ASM86 REPORT1
DIR REPORT1.*
ERA *.BAK
PIP PRN:=REPORT1.PRN
ERA REPORT1.PRN
```

SUBMIT creates the $$$.SUB file on drive A. SUBMIT then executes the $$$.SUB file. When SUBMIT executes the file, drive A is automatically logged. Command processing is terminated if a system error occurs during execution of a $$$.SUB file.

15.7 THE USER (DISPLAY AND SET USER NUMBER) COMMAND

The USER resident command displays or changes the current user number. To display your current user number, enter --

```
user(cr)
```

You can divide the disk directory into distinct groups according to user numbers. Files you create under one user number are inaccessible under other User numbers. However, you can make user Ø files accessible to other user numbers with the STAT [SYS] command. You can move files between user areas using PIP with the G parameter.

When you start CP/M-86 the current user number is Ø; the user number will remain Ø unless you change it with the USER command.

To change the current user number, use the command --

```
     user n(cr)
```

where n is a user number in the range Ø to 15.

You can also use the USER command to find out what user number a file was created under. To display a list of user numbers that have files associated with them, enter the command --

```
     stat usr:(cr)
```

The following example illustrates displaying the current user number with USER.

```
     A>user(cr)
     Ø
     A>
```

The next example demonstrates changing the current user number to 3.

```
     A>user 3(cr)
```

If you are in User 3 and want to get a file named X from User 7, use PIP as in the following example. The example assumes that PIP.CMD is a Sys file in User Ø's directory, or is in User 3's directory.

```
A>pip y=x[g7] (cr)
```

16. ED: THE CP/M-86 TEXT EDITOR

Use the CP/M-86 text editor (ED) to create or modify ASCII files. ASCII files can be programs, documents, or data for programs.

ED creates files from text entered at the keyboard. When used to modify an existing file, ED reads the file to be edited into a portion of memory called the edit buffer and modifies it according to operator commands. The edit buffer is empty until the operator loads it with either new text or text from an existing file.

During the edit, the changed file can be written to a temporary file with the same file name as the original file, but with the extension of .$$$. After the edit, ED renames the temporary file to the original file name. ED keeps the original file and assigns it the extension .BAK, as a backup file.

For example, to modify the existing file, SAMPLE.TXT, ED creates a temporary file named SAMPLE.$$$ to be used during the ED session to contain the newly modified file. When the operator terminates the session, ED renames the original file as SAMPLE.BAK and renames SAMPLE.$$$, the temporary file, SAMPLE.TXT. Refer to Section 16.7 for a sample ED session.

16.1 ACTIVATING THE TEXT EDITOR

To activate the CP/M-86 text editor, enter the ED command after the system prompt. The general format of the ED command follows--

ed filespecifier [filespecifier](cr)

where ED is the command stem and the first filespecifier is the name of the new or existing source file.

The second file specifier is the optional drive and/or name under which to save an edited file. If a second file specifier is not included, the edited file will be saved using the name of the first file specifier.

If no file with the given source file specifier exists, ED creates a new file with that file specifier, as described in 16.3. If you omit the file specifier in the ED command line, the system responds with the error message -

```
DISK OR DIRECTORY FULL
```

When activated, ED displays its command prompt, an asterisk (*).

16.2 BASIC OPERATING PRINCIPLES

While inputting at the keyboard, terminate each line with a carriage return (a line feed is supplied automatically with each carriage return).

ED assigns and displays line numbers for each line of text for use during the editing session. The line numbers are not written into the file. Use the line numbers to refer to specific lines of text that are to be modified.

ED commands should be entered as lower case characters, even though ED accepts both upper and lower case. If the editing commands I, F, S, N, J, M, and R are entered in upper case, subsequent text will be translated to upper case, regardless of how it is entered or displayed on the screen.

Save in-process work at regular intervals. If data is accidentally deleted from the edit buffer, this step will minimize the loss.

Before starting an editing session, use the STAT command to check the amount of free space on your disk. Make sure that the unused portion of your disk is at least as large as the file you are editing (larger if you plan to add characters to the file). When ED finds a disk or directory full, ED has only limited recovery mechanisms.

16.3 CREATING A TEXT FILE

To create a text file, after the system prompt (>) enter the ED command and the file specifier you want to assign to the new file:

```
     ed filespecifier(cr)
```

ED responds with the message NEW FILE when you enter an ED command naming a file specifier that is not already on disk. When ED is ready to accept a command, the screen displays the ED command prompt, an asterisk (*).

To store the results of an editing session on another drive or in another file, include a destination drive name and/or destination file name in your ED command.

The following example demonstrates opening the file MYFILE.TXT on drive A and sending the edited material to MYFILE.TXT on drive B.

```
     A>ed myfile.txt b:(cr)
```

The next example illustrates using the ED command to edit the file MYFILE.TXT and to store the resulting edit in YOURFILE.TXT.

```
A>ed myfile.txt yourfile.txt(cr)
```

If the destination file that you name already exists, ED prints the following error message and terminates --

```
Output File Exists, Erase It
```

Prepare ED to accept text input by entering the i (insert) command:

```
i(cr)
```

ED indicates it is ready to accept input by displaying the first line number. The following example illustrates opening a new text file named JOE.TXT.

```
A>ed joe.txt(cr)

    NEW FILE
         : *i(cr)
        1:
```

To terminate text insertion, enter ALT-Z. ED responds to an ALT-Z by displaying its prompt, an asterisk (*).

16.4 EDITING AN EXISTING FILE

To edit an existing file, enter the file's specifier, as well as destination file specifier if desired, with the ED command. After the ED command prompt, load the file to be edited into the edit buffer by entering the A (append) command:

```
na(cr)
```

where n is a number entered before the A command to indicate the number of lines of the file to be loaded into the edit buffer. A pound sign (#) entered in the preceding command represents the number 65535, the largest number of lines that ED will allow; enter the pound sign before the A command to load the entire file into the edit buffer. If no integer is included in the A command, ED assumes the number 1 (see Table 16-1).

In the following example, the file SAMPLE.TXT is loaded into the edit buffer for modification.

```
     A>ed sample.txt(cr)

               : *#a(cr)

              1: *
```

Use of the A command is explained in detail in Table 16-1.

16.5 TEXT TRANSFERS BETWEEN THE EDIT BUFFER AND ASCII FILES

The edit buffer is a text storage area in the computer memory. Editing commands affect text in the edit buffer; they do not affect files on the diskette. Table 16-1 describes the commands that move lines of text between the edit buffer and the temporary file (with given extension .$$$) created by ED to store work during an editing session. Figure 16-1 illustrates use of these commands.

Table 16-1: **Edit Buffer - Text File Transfer Commands**

COMMAND FORMAT	COMMAND FUNCTION
nA(cr)	Appends to the last character in the edit buffer the next n lines from original text file to be edited. Entering #A loads 65535 lines of text or the entire file, whichever is smaller, into the edit buffer from the original text file.
nW(cr)	Writes the first n lines of the edit buffer into temporary file. Remaining lines are moved up to the top of buffer.
E(cr)	Ends edit. Edit buffer is copied into temporary file, and files are renamed.
H(cr)	Allows work accomplished so far to be saved. Buffer is emptied, temporary file becomes new source file, and a new temporary file is created (equivalent to entering an E command, followed by the ED command with the file specifier for the source file).
O(cr)	Returns to original file. Edit buffer is emptied, temporary file is deleted, and edit starts over again on original file.

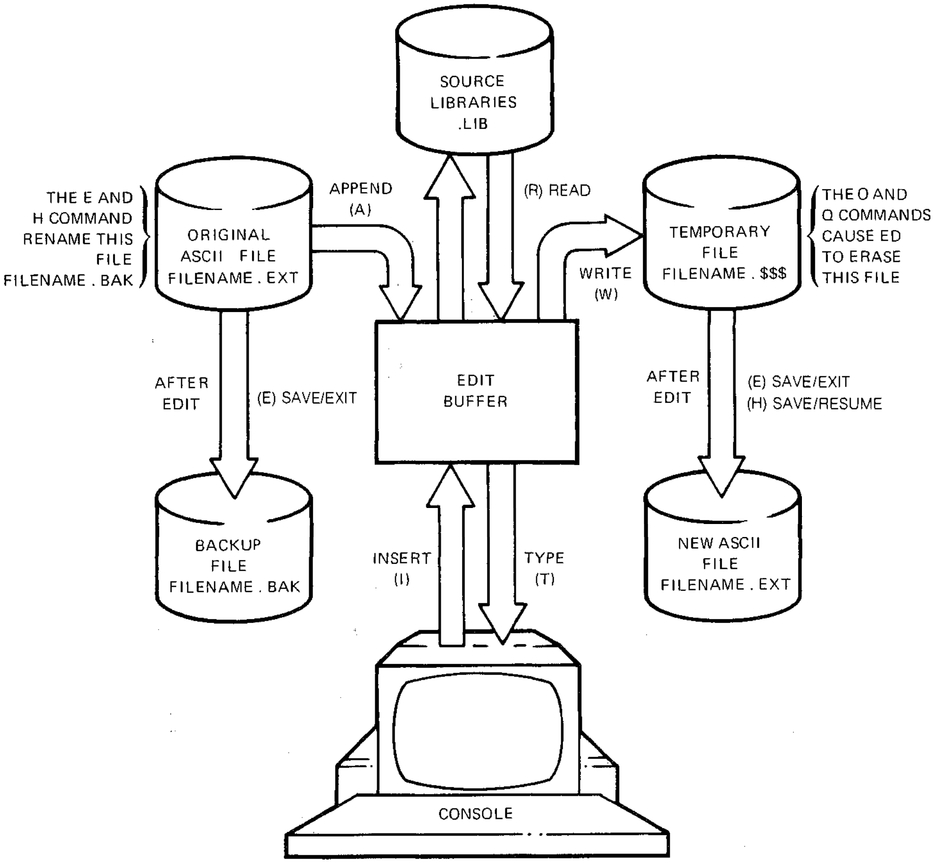
COMMAND FORMAT	COMMAND FUNCTION
Q(cr)	Quits the edit with no file changes, erases the .BAK file if one existed, and returns control to the operating system.
rfilespecifier (cr)*	Reads specified file (whose extension must be .LIB) into the edit buffer. If the file specifier is omitted, X$$$$$$$.LIB is read in. Input file is left intact. This command is useful in inserting boilerplate information into more than one file.
nX(cr)	Transfers n lines of data from edit buffer to temporary file to which ED assigns the name X$$$$$$$.LIB.

NOTES: Many ED commands can be preceded by an integer represented in the command format column by n. If no integer is entered, ED assigns a value of 1. If a pound sign (#) is entered for n, 65535 (the largest allowable integer) is assigned.

Commands can be entered in either upper or lower case, unless otherwise noted.

* To avoid translation of all text following an R command into upper case, enter the R command in lower case (see 16.2).

Figure 16-1: Edit Buffer/Edit File Interaction


SOURCE LIBRARIES .LIB
THE E AND H COMMAND RENAME THIS FILE FILENAME . BAK
ORIGINAL ASCII FILE FILENAME . EXT
APPEND (A)
(R) READ
WRITE (W)
TEMPORARY FILE FILENAME . $$$
THE O AND Q COMMANDS CAUSE ED TO ERASE THIS FILE
AFTER EDIT
(E) SAVE/EXIT
EDIT BUFFER
AFTER EDIT
(E) SAVE/EXIT
(H) SAVE/RESUME
BACKUP FILE FILENAME . BAK
INSERT (I)
TYPE (T)
NEW ASCII FILE FILENAME . EXT
CONSOLE

16.6 THE CHARACTER POINTER

Once ASCII characters are in the edit buffer, ED creates a character pointer (CP) in the buffer. The CP, not visible on the screen, is moved throughout the edit buffer under operator control. The location of the CP determines where the commands perform their tasks. The line that contains the CP is called the current line (CL).

The CP points to the imaginary space between two characters (spaces are considered characters); it never points to the character itself. The CP can also point to the beginning of the edit buffer (before the first character) or to the end of the buffer (the end of the last active character).

Always position the CP before editing text, or data may be accidentally lost or modified.

The commands that manipulate the CP or display text in the vicinity of the CP are described in Table 16-2. In the table, an n preceding a command indicates that an optional value can be specified. If no value is specified, ED assigns a value of 1. If the pound sign (#) is entered for n, ED assigns a value of 65535. In the table, commands may be entered preceded by either a plus (+) or a minus (-) sign. If neither sign is entered, ED assigns a +.

Table 16-2: Character-Pointer Related Commands

COMMAND FORMAT	COMMAND FUNCTION
±B(cr)	Moves CP to beginning (+) or end (-) of the edit buffer.
±nC(cr)	Moves CP n characters toward the beginning (-) or end (+) of buffer. (Carriage return and line feed are counted as two characters.)
±nL(cr)	If n equals Ø, moves CP to beginning of CL. If n does not equal Ø, moves CP to beginning of CL, then up (-) or down (+) n lines. CP stops at top or bottom of edit buffer if n is too large.
±nT(cr)	If n equals Ø, displays CL up to CP. If n equals 1, displays CL from CP to end of the line. If n is greater than 1, displays CL and the following n-1 lines (+) or the preceding n lines (-). This command does not move the CP. (To display the CL, enter ØTT.)
±n(cr)	Moves CP as does the L command, and then displays new CL.
n:(cr)	Moves CP to beginning of line number n.
:n(cr)	Defines a range of text between the current line and line number n. This does not move the CP.

16.7 MODIFYING TEXT

Once the CP has been positioned, text in the edit buffer can be modified. Single characters or complete lines of text can be deleted. Text from a different ASCII file can be added, or new data can be inserted. Existing strings -- groups of contiguous characters -- can be replaced with new strings. (Strings are limited to a maximum of 100 characters; a string of 0-length, called a null string, is allowed.) The commands used to modify text are described in Table 16-3.

Table 16-3: TEXT MODIFICATION COMMANDS

COMMAND FORMAT	COMMAND FUNCTION
±nD	Deletes n characters preceding (-) or following (+) CP.
±nK	Deletes n lines preceding (-) or following (+) CL. If CP is not at beginning of line, CL characters preceding (+) or following (-) CP are not deleted.
nfstring or nfstring^Z	Finds specified string. Use ALT-Z if additional commands will be entered in the same line. Search begins at CP. Match is made n times, and CP is positioned after last character in last matched string. If n matches cannot be made, CP remains at its initial position. String can include ALT-L, which matches a carriage return/line feed in a string.

COMMAND FORMAT	COMMAND FUNCTION
jstring1^Z string2^Z string3 or njstring1^Z string2^Z string3^Z	Text between string 1 and string 3 is replaced with string 2. Searches for string 1, inserts string 2, places CP after string 2, and deletes original characters between string 1 and string 3. Search begins at the CP, and substitution is made n times. If string 3 is not found, nothing is deleted.
nmstring or nmstring^Z	Allows more than one command to be entered and executed repeatedly. String represents a string of commands, rather than text. If n is greater than 1, command string is executed n times. If n equals Ø or 1, command string is executed repeatedly until an error condition is encountered (e.g., end of buffer is reached with F command). For example, the command string MSGAMMA^ZDELTA^ZOTT(cr) substitutes DELTA for each GAMMA in the buffer, and displays each changed line.

COMMAND FORMAT	COMMAND FUNCTION
nnstring	Finds specified string. Search begins at current position of CP. If string cannot be found in edit buffer n times, contents of buffer are written to temporary file (automatic #W command). Input lines from original file are read into edit buffer (automatic A command) until buffer is half full or original file has been completely transferred. Search continues in this manner until string has been found n times (or original file has been completely transferred). This command is used when entire original file cannot fit in edit buffer.
istring or istring^Z	Inserts string in front of the CP. (A carriage return without ALT-Z is entered after string to put a carriage return/line feed at the end.)
nsstring1^Z string2 or nsstring1^Z string2^Z	Searches for string 1 and substitutes string 2 for string 1 n times. Search begins at current position of CP. Search is limited to contents of edit buffer.
Rstring	Inserts file into edit buffer before CP. String represents a file specifier whose extension is assumed to be .LIB. (i.e., the command RMACRO(cr) inserts MACRO.LIB file into buffer after CP).

16.8 COMMAND STRING USAGE

Two or more commands can be entered as a string (up to 128 characters); as in the case of single command entries, ED executes them only after a carriage return is entered. Table 16-4 lists the CP/M-86 ALT characters that manipulate the input command string.

Table 16-4: Command String Alternate Characters

ALTERNATE CHARACTER	FUNCTION
ALT-H or Backspace	Deletes the preceding character.
ALT-X	Deletes the entire command string.
ALT-C	Returns control to the operating system without saving any of the text modifications that are in the edit buffer.
ALT-E	Returns carriage without transmitting command line (128 characters maximum).

16.9 SAMPLE ED SESSION

The following example illustrates use of some of the basic ED functions described in this chapter. Brief explanations of the user entries are shown on the right.

```
A>ed sample.txt(cr)       Edit a file called
                           SAMPLE.TXT.
NEW FILE
  : *i(cr)                 Insert text into edit
                           buffer.
 1:  This is a simple(cr)
 2:  file to be used(cr)
 3:  to demonsttrate(cr)
 4:  how to use(cr)
 5:  a program called(cr)
 6:  ED.(cr)
 7:  ^Z                    Terminate text insertion.
  : *b(cr)                 Move CP to beginning of
                           buffer.
 1: *#t(cr)                Display entire buffer.
 1:  This is a simple
 2:  file to be used
 3:  to demonsttrate
 4:  how to use
 5:  a program called
 6:  ED.
 1: *ssim^Zsam^ZØtt(cr)    Substitute SAM for SIM,
                           changing simple to sample,
                           and then display corrected
                           line.

 1:  This is a sample
 1: *3:stt^Zt^ZØtt(cr)     Move CP to line 3 and
                           change tt to t and then
                           display corrected line.

 3:  to demonstrate
 3: *5:kb#t(cr)            Move CP to line 5 and
                           delete it, then move CP
                           to beginning of buffer and
                           display entire contents.
```

```
1:   This is a sample
2:   file to be used
3:   to demonstrate
4:   how to use
5:   ED.
1: *5:i(cr)                Move CP to line 5 and
                           insert new text.

5:   the text Editor
6:   known as
7:   ^Z
7: *3:#t(cr)               Move CP to line 3 and
                           display the rest of the
                           buffer.

3:   to demonstrate
4:   how to use
5:   the text Editor
6:   known as
7:   ED.
3: *3c11dishow^ZØtt(cr)     Move CP 3 columns, delete
                            next 11 characters, insert
                            "show," display new line.

3:   to show
3: *b3t(cr)                Move CP to beginning of
                           buffer and display 3 lines.

1:   This is a sample
2:   file to be used
3:   to show
1: *2:i(cr)                Move CP to line 2 and
                           insert text.

2:   ASCII text(cr)
3:   ^Z
3: *b#t(cr)                Move CP to beginning and
                           display entire buffer.
```

```
 1:  This is a sample
 2:  ASCII text
 3:  file to be used
 4:  to show
 5:  how to use
 6:  the text Editor
 7:  known as
 8:  ED.
 1: *e(cr)                   Exit ED. Save text in file
                             SAMPLE.TXT.
A>
```

16.1Ø ED ERROR MESSAGE AND ERROR INDICATORS

When errors occur, ED displays the last character read before the error and an error indicator. The following line shows the error message format:

BREAK errorindicator AT lastcharacter

Table 16-5 describes the ED error indicators.

Table 16-5: ED Error Indicators

ERROR INDICATOR	MEANING
?	Command is not recognized.
>	Memory buffer full (use D, K, N, S, or W command to remove characters); F, N, or S strings are too long.
#	Command cannot be executed number of times specified.
O	LIB file cannot be found.
E	Command aborted.
F	File error.

16.11 ALTERNATE-CHARACTER SUMMARY

Table 16-6 summarizes the alternate characters available in ED.

Table 16-6: Text Editor Alternate-Character Commands

ALTERNATE CHARACTER	FUNCTION
ALT-C	Restart (use with extreme care -- loss of data may occur).
ALT-E	Physical carriage return (not actually entered in command) which allows commands longer than one line to be entered at console.
ALT-H	Character delete (same as Backspace key).
ALT-I or Tab Key	Logical tab (columns 1, 8, 15, etc.).
ALT-L	Logical carriage return/line feed for use in search and substitute strings.
ALT-Z	String terminator.

16.12 ED COMMAND SUMMARY

Table 16-7 summarizes the ED commands in an alphabetical listing.

Table 16-7: Alphabetical List Of ED Commands

COMMAND	FUNCTION
nA	Appends lines.
±B	Moves CP to beginning of buffer.
±nC	Moves CP by characters.
±nD	Deletes characters.
E	Ends edit and closes file (normal end).
nf	Finds string in edit buffer.
H	Ends edit; closes and reopens file.
i	Inserts characters.
nj	Places string in juxtaposition (between specified strings).
±nK	Deletes lines.
±nL	Moves CP by lines.
nm	Enters and executes more than one command at a time.
nn	Finds characters throughout original file.

COMMAND	FUNCTION
O	Returns to original file.
nP	Displays next 23 lines (1 screen) in buffer.
Q	Quits edit with no file changes.
R	Reads library file into buffer.
nS	Substitutes one string for another.
±nT	Displays lines.
nW	Writes lines into temporary file.
nX	Transfers text from edit buffer to temporary library file.*
±n	Moves CP and displays (±nLT).
n:	Moves CP to line n.
±V	Enables/disables line numbering.

*ØX filespecifier allows you to erase a file with ED. This is useful if the disk or directory is full.

SECTION IV

HARD DISK UTILITIES

17. THE AUTOSET UTILITY

The AUTOSET utility allows you to configure the hard disk (prepare it for use) quickly and easily. AUTOSET initializes the hard disk and divides it into volumes. You can use AUTOSET to configure a new hard disk or a hard disk that has already been used, or initialized.

17.1 AN OVERVIEW OF AUTOSET

After you load AUTOSET, it displays information about your configuration options. Each option is designed for a specific type of use. For example, one option divides the hard disk into two volumes, appropriate for use with a single application that creates large files.

Configuration data for each option is stored in a file. When you have viewed the information about your options, AUTOSET asks you to choose one by typing its file name. AUTOSET then uses the information in the file to configure your hard disk.

If you already know the name of the file containing the configuration data you need, you can type the file name with the AUTOSET command when you load the utility. If you do this, AUTOSET configures the hard disk directly without displaying information about the options.

The configuration files available with AUTOSET are described later in this chapter. You can also create your own configuration file by using the utility HDSETUP. HDSETUP is described in the **Hard Disk Programmer's Tool Kit**, available from your dealer. Configuration files must have the extension .CFG.

17.2 LOADING AUTOSET

Load AUTOSET by typing:

```
autoset(cr)
```

You can also load AUTOSET by typing:

```
autoset filename(cr)
```

where filename is the name of the configuration file (you don't need to type the extension).

When loaded AUTOSET displays its sign-on message:

```
AUTOMATIC SETUP UTILITY v1.Ø
```

If your hard disk is already initialized, AUTOSET also displays the message:

```
Your hard disk is already initialized.
All of the current data will be destroyed if
this program continues.
Do you want to back up your data? (y/n)
```

You must type N to continue with the configuration. If you type any other character the program terminates. If you type N, AUTOSET asks:

```
Are you sure you want to reinitialize the disk?
(y/n)
```

If you type Y, the program continues. If you type any other character, the program terminates.

If you typed a file name with the AUTOSET command, AUTOSET configures the hard disk and displays the message:

Press any key to reboot your system.

Type any key to reboot the DOS from the floppy diskette. You can then use the SYSCOPY utility to copy the DOS onto the hard disk boot volume. The SYSCOPY utility is described in this guide in the section on MS-DOS. The HELP file, described next, shows you which volume is the boot volume for each configuration option. After you copy the DOS onto the hard disk, you can load the DOS directly from the hard disk.

17.3 THE HELP FILE

The file AUTOSET.HLP contains information about the configuration files available with AUTOSET. If you type the AUTOSET command without a file name, this information is displayed. (AUTOSET.HLP must be on the default drive for the information to be displayed.)

The information in the help file is divided into pages. At the end of each page, this message appears:

Hit any key to continue.

Hit any key to move to the next page of information.

After you have viewed the last page, AUTOSET lists the names of the AUTOSET configuration files and the message:

Please enter the file name of your choice.

Type one of the displayed file names, or the name of a configuration file prepared with HDSETUP. You can also terminate the program by typing ALT-C or any other file name.

After you type the file name, AUTOSET configures the hard disk and displays the message:

Press any key to reboot your system.

Press any key to reload the DOS from the floppy diskette. You can then use the newly configured hard disk.

17.4 AUTOSET CONFIGURATION OPTIONS

Descriptions of the configurations available with AUTOSET follow. The configurations are listed by their file name. Specifications for the configuration appear at the beginning of each description. These specifications also appear in the AUTOSET help file.

For each configuration, the specifications show:

- The number of volumes created.

- The capacity of each volume in K bytes (1K = 1Ø24 bytes).

- The number of files that can be stored under each volume's directory.

- o The allocation unit (AU) size. (This information is used by programmers.)

- o The amount of DOS memory used by each volume. The total amount of DOS memory required for the configuration is also shown at the bottom of the specification information.

To the right, the drive letter assignments are shown. The volume used to load the DOS is labeled "boot." The floppy disk drive assignments are labeled "floppy."

5AND5.CFG CONFIGURATION

Specifications				A: 5m (boot)
Capac	Dir	Au	Mem Req	B: Floppy
5000K	352	4k	2048b	C: 5+m
5000+k	352	4k	2048b	

Total Memory Requirements with double-sided floppy-
5k = 5120b

This configuration is appropriate if you use 2 applications only. You can assign one volume to each application. The configuration handles large or small files easily and efficiently.

This configuration is not appropriate if you use many applications.

2AND3.CFG CONFIGURATION

Specifications

Capac	Dir	Au	Mem Req
2000K	256	4k	1024b
3000+k	512	4k	3584b

A:	2m (boot)
B:	Floppy
C:	8+m

Total Memory Requirements with double-sided floppy- 4.5k=4608b

This configuration is appropriate if you use 1 application that creates large files - such as an accounting spread sheet application. You can use the smaller volume for the application and the larger volume for data storage.

This configuration is not appropriate if you use many applications. The large data volume does not allow you to group many small files logically.

1AND3X3.CFG CONFIGURATION

Specifications

Capac	Dir	Au	Mem Req
1000K	192	2K	1024b
3000K	352	4K	1536b
3000K	352	4K	1536b
3000+K	352	4K	1536b

A:	1m (boot)
B:	Floppy
C:	3m
D:	3m
E:	3+m

Total Memory Requirements with double-sided floppy - 6.5K = 6656b

This configuration is appropriate if you use up to 3 applications. It is well suited for storing small files and can handle a moderate number of large files. Very small files (less than 2K bytes) should be stored on the drive A volume (the boot volume) because of its small AU size.

A disadvantage of this configuration is that it uses a large amount of DOS memory. Memory sensitive applications may have problems.

5X2.CFG CONFIGURATION

Specifications

Capac	Dir	Au	Mem Req
2000K	256	2K	1536b
2000K	256	2K	1536b
2000K	256	2K	1536b
2000K	256	4K	1024b
2000+K	256	4K	1024b

A:	2m (boot)
B:	Floppy
C:	2m
D:	2m
E:	2m
F:	2+m

Total Memory Requirements with double-sided floppy - 7.5K = 7680b

This is a good general purpose configuration. Small files are efficiently stored on the first 3 volumes because of the small AU size. Larger files should be stored on the remaining 2 volumes. This configuration is your best choice if you use a word processor to create memo-size files.

A disadvantage of this configuration is that it uses a large amount of DOS memory. However, this should not be a problem if you work primarily with small files.

2AND8X1.CFG CONFIGURATION

Specifications

Capac	Dir	Au	Mem Req
2000K	256	4K	1024b
1000K	240	2K	1024b
1000K	240	2K	1024b
1000K	240	2K	1024b
1000K	240	2K	1024b
1000K	240	2K	1024b
1000K	240	2K	1024b
1000K	240	2K	1024b
1000+K	240	2K	1024b

A:	2m (boot)
B:	Floppy
C:	1m
D:	1m
E:	1m
F:	1m
G:	1m
H:	1m
I:	1m
J:	1+m

Total Memory Requirements with double-sided floppy - 10K = 10240b

This configuration is designed for the programmer. It allows you to store a variety of files on different volumes. This promotes good file organization.

18. THE VDIR UTILITY

The VDIR utility displays information about the drive assignments on your computer. VDIR lists information about each volume or floppy disk drive assigned to a drive letter.

Before you use VDIR, check that your floppy disk drive(s) contain MS-DOS diskette(s). Also, if you use an external hard disk, check that its power is on. Otherwise, VDIR cannot list drive information for the drive letters assigned to the floppy drives or the external hard disk. Instead, VDIR displays an MS-DOS BIOS error message and the message NO DISK (see the second example below).

VDIR lists the drive information under headings:

- VOLUME NAME lists the volume name. If a floppy disk drive is assigned to the drive letter and the drive contains an MS-DOS diskette, it lists the name FLOPPY.

- VOLUME TYPE lists the DOS used with the volume or the diskette in the floppy disk drive. (Currently, MS-DOS is the only DOS listed.)

- VOLUME CAPACITY lists the capacity of the volume or diskette in the floppy disk drive. The capacity is listed in K bytes (1K equals 1024 bytes).

- VOLUME DIRECTORY ENTRIES lists the number of files you can store under the volume or diskette directory.

- o VOLUME ALLOCATION UNIT lists the size of the allocation unit for the volume or diskette.

The following example shows using VDIR on a computer with an internal hard disk containing two volumes:

```
A>vdir(cr)

VOLUME DIRECTORY UTILITY  V1.Ø

                                          DIRECTORY
 VOLUME NAME      TYPE      CAPACITY       SIZE        AU

A:   VOLUME 1    MS-DOS       2ØØØK         256        4K
B:   FLOPPY      MS-DOS        612K         128        2K
C:   VOLUME 2    MS-DOS       8327K         512        4K

A>
```

The following example shows using VDIR on the same computer when the floppy disk drive is empty:

```
A>vdir(cr)

VOLUME DIRECTORY UTILITY V1.Ø

Drive= 1, Track= Ø, Sector= Ø, Error= FA

                                              DIRECTORY
 VOLUME NAME          TYPE      CAPACITY        SIZE      AU

A:   VOLUME 1        MS-DOS       2ØØØK          256      4K
B:   NO DISK
C:   VOLUME 2        MS-DOS       8327K          512      4K

A>
```

19. THE ARCHIVE UTILITY

Use the ARCHIVE utility to back up hard disk files. ARCHIVE saves hard disk files by recording them on diskette. Saved files may be restored by copying them from diskette back to the hard disk with the RESTORE command.

Files can be saved and restored singly or in groups. Saved files are written to one or more diskettes, depending on the number and size of the files. Large files can be recorded across several diskettes. Restored files are read from diskette to a designated hard disk volume.

19.1 AN OVERVIEW OF ARCHIVE

The two basic ARCHIVE commands are ARCHIVE and RESTORE. With either command, the utility prompts you through each step of the process.

The ARCHIVE command writes the files you specify on diskette(s). These diskettes are called the **Archive Diskette Library.** Each diskette contains an **Archive Directory** that records which files are contained on which diskette or diskettes.

The RESTORE command writes the files in the Archive Diskette Library back to a hard disk volume. During a RESTORE, you are prompted to insert specific diskettes from the Archive Diskette Library.

File groups are saved or restored in wild-card groups or by **argument file.** An argument file is a file that contains a list of files to be archived; the files are listed one per line. Up to 415 files can be archived in a run.

You can quit the utility at any point by typing **Q(cr).** If you quit during the middle of an ARCHIVE or RESTORE command (after having completed at least one diskette), you can restart where you left off by re-entering the command line.

The following sections describe the ARCHIVE commands. In the command syntax descriptions:

- o Uppercase words or letters are to be typed exactly as shown.
- o Square brackets ([]) enclose optional parameters.

19.2 THE ARCHIVE COMMAND

19.2.1 COMMAND SYNTAX

The ARCHIVE command syntax is:

> **ARCHIVE [&]filespec [SINCE date]**
> **[UNDER archivename] FROM volume TO**
> **diskettedrive [DOUBLE](cr)**

Where:

& Indicates that filespec is an argument file specifier.

filespec Is a file specifier -- with or without wild-cards -- that identifies the files to be archived. If it is an argument file specifier, it must contain the drive designation, file name, and file extension -- and can include wild-cards.

date Specifies that only files modified since this date are to be archived. Format for typing the date is mm/dd/yy, or dd-mm-yy. (The date is displayed as mm/dd/yy.) The default date is 1-1-80.

archivename Is the name assigned to the archive library. The default is the current date plus the extension .ARC - for example, 1-20-83.ARC.

volume Is the letter name (A - P) of the hard disk volume from which the files are to be archived.

diskettedrive Is the letter name (A - P) of the floppy disk drive to be used.

double Specifies that double-sided diskettes are to be used. The default is for single-sided diskettes.

19.2.2 ARGUMENT FILES

An argument file contains a list of the files you want to archive. Create an argument file by listing the files to be archived one on a line. An argument file can contain up to 415 entries. Each entry must give the file's drive designation, file name, and file extension.

The file specifier for an argument file must contain a drive designation, file name, and file extension.

19.2.3 COMMAND LINE EXAMPLES

The following command line archives all files modified since 9/1/82 from hard disk volume K. The archive library is named SEPT.ARC, and the archived files are written to double-sided diskette(s) in floppy drive B:

```
     archive *.* since 9/1/82 under sept.arc from k
     to b double(cr)
```

The following command line uses default values. The & tells you K:FILE.LST is an argument file. The files listed in K:FILE.LST are to be written from volume K to single-sided diskette(s) in floppy drive A. The archive library name is (by default) the current date with the .ARC extension. All files modified since 1/1/8Ø (the default date) are archived:

```
     archive &k:file.lst from k to a(cr)
```

19.2.4 COMMAND EXAMPLE

The following example shows how to use ARCHIVE. In the example, what you type is underlined.

```
C>archive *.* since 1Ø/1/82 under oct.arc from c to
d(cr)

Archive
Version 1.x

Archiving files modified since 1Ø/Ø1/82 under oct.arc on
single sided diskettes.

Insert a newly formatted diskette in drive d.

Press return when ready.  (Q to quit program)(cr)

Writing:
     Diskette 1 of 15.

Writing to diskette 1 complete.

Now

     1) Remove diskette 1.
     2) Label it:
          oct.arc
          Diskette 1 of 15
          10/05/82

     3) AFFIX A WRITE PROTECT TAG TO THE DISKETTE

You will need 14 additional newly formatted diskettes to
archive the rest of the files.

Do you need to format diskettes? (y/n)y

C>
```

In this example, ARCHIVE has:

1) Created the Archive Directory (a list of the files to be archived) and allocated files to the diskettes in the Archive Diskette Library.

2) Transferred the file(s) allocated to diskette 1 onto the first diskette inserted.

3) Written 2 utility files to diskette 1: ARC.DAT and ARC.ALF. ARC.DAT contains the Archive Directory. ARC.ALF contains housekeeping information required by ARCHIVE and RESTORE. These utility files are written to each archive diskette after the diskette is processed.

Diskette 1 - which has been fully processed - can be used to restart the interrupted archive after more diskettes have been formatted.

19.3 THE RESTORE COMMAND

The command line syntax for the RESTORE command is:

RESTORE [&]filespec FROM diskettedrive TO volume(cr)

The command parameters are the same as for ARCHIVE. With the RESTORE command, the specified files are recorded from diskette(s) in the specified floppy disk drive onto the specified hard disk volume.

Any diskette from the Archive Diskette Library can be used to start a RESTORE. The **Restore Directory** (a list of the files to be restored) is constructed from the files on the inserted diskette. A copy of the Restore Directory is saved in a temporary file (ARC.$$$) on the target restore volume. You must restore files on a volume sufficient to hold the restored files and ARC.$$$. (ARC.$$$ requires 1 to 2 allocation units.)

The following example shows how to use RESTORE. In the example, what you type is underlined.

```
F>Restore *.* from b to f(cr)

Restore
Version 1.x

Restoring from b to f.

Insert a diskette from the archive library into
drive b. Press return when ready. (Q to quit
program) (cr)
```

```
Archive Library oct.arc
This diskette is number 7 of 15.
This library contains files of volume c
archived on 10/05/82.

Insert diskette 1 in drive b.
Press return when ready. (Q to quit program)(cr)

Archive Library oct.arc
This diskette is number 1 of 15.

Restoring:

     filename1
     filename2
         .
         .
Restoration of diskette 1 complete.

Insert diskette 2 in drive b.
Press return when ready. (Q to quit program)(cr)

Archive Library oct.arc
This diskette is number 2 of 15.

[and so on....]

Restoring complete.

F>
```

19.4 RESTARTING AN INTERRUPTED ARCHIVE OR RESTORE COMMAND

You can restart an interrupted ARCHIVE or RESTORE run with the following commands:

```
    ARCHIVE TO diskettedrive(cr)

         -or-

    RESTORE [FROM diskettedrive] TO volume(cr)
```

You must complete at least one diskette with the original command to restart an interrupted ARCHIVE or RESTORE. If you use a floppy drive other than the original floppy drive, make sure it can handle the capacities of the original command.

When you restart an ARCHIVE or RESTORE command, check that you insert the correct diskette. If you insert the wrong diskette, type a Q to exit the program and retype the restart command line.

The following example shows how to restart an interrupted ARCHIVE command:

```
C>Archive to d(cr)

Archive
Version 1.x

Continuing to d.

Insert the last completed Archive diskette into
drive d.  Press return when ready. (Q to quit
program) (cr)

Archive Library oct.arc
This diskette is number 1 of 15.
This library contains files of volume c
archived on 1Ø/Ø5/82.
```

```
Insert a newly formatted diskette into drive d.
Press return when ready. (Q to quit program)(cr)

Writing:

     Diskette 2 of 15.

Writing to diskette 2 complete.

[...and so on]

Archiving complete.

C>_
```

The following example shows how to restart a RESTORE command. In the example, RESTORE is interrupted after the second diskette has been processed:

```
K>Restore *.* from n to k(cr)

Restore
Version 1.x

Restoring from n to k.

Insert a diskette from the archive library into
drive n. Press return when ready. (Q to quit
program) (cr)

Archive Library total.arc
This diskette is number 7 of 15.
This library contains files of volume c
archived on 9/11/82.

Insert diskette 1 in drive n.
Press return when ready. (Q to quit program)q
```

```
K>Restore to k(cr)

Restore
Version 1.x

Continuing from n to k.


Insert a diskette from the Archive Library oct.arc
into drive n. Press return when ready. (Q to quit
program) (cr)

Archive Library oct.arc
This diskette is number 2 of 1Ø.

Insert diskette 1 in drive n.
Press return when ready.  (Q to quit program) (cr)

Archive Library oct.arc
This diskette is number 1 of 1Ø.

[and so on....]

Restoring complete.

K>
```

2Ø. OPERATOR UTILITIES ERROR MESSAGES

2Ø.1 AUTOSET ERROR MESSAGES

Following are the errors you may encounter while using the AUTOSET utility. The error messages occur before configuration takes place. If these errors do occur, the hard disk is left in the original state.

- **The AUTOSET.HLP file was not found.**
  The help file could not be found on the default drive. The utility terminates.

- **Unable to close the AUTOSET.HLP file.**
  The help file could not be closed. The utility terminates.

2Ø.2 HARDWARE ERROR MESSAGES

The following error messages may be caused by hardware. Try reloading the operating system.

- **Cannot open the specified file name.**

- **Error reading input file.**

- **Cannot close input file.**

2Ø.3 HARD DISK ERROR MESSAGES

Hard disk errors can also occur. If a hard disk error occurs, you are notified by an error message. If the error persists, try resetting the system by slowly turning the power off and on. If this does not help, contact your dealer to have the hardware and media checked.

Following are the messages you may encounter while using the hard disk:

- **Disk reading error.**
- **Memory error.**
- **Error writing FATS and directory.**
- **Error writing new volume label with assignments.**
- **Error writing new volume label.**
- **Error writing volume label with new assignments.**
- **Error writing newly freed volume label.**
- **Error writing valid drive label.**
- **Error writing drive label backup.**
- **Controller error. Unable to reset hard disk.**

2Ø.4 ARCHIVE ERROR MESSAGES

Following are the error messages you may encounter while using the ARCHIVE utility.

- **Open <filename>, cannot be closed.**

  Check that current diskette is correct. Press Return when ready (Q for emergency exit).

- **Hard disk full; <filename> cannot be written.**

  Start over using a hard disk volume with more space. Program terminates.

- **Volume <X> not available.**

  Program terminates.

- **Volume <X> is available but is empty.**

  Program terminates.

- **<X> is not an available floppy drive.**

  Program terminates.

- **Improper response.**

  Prompt is repeated.

- **Improper syntax for an archive name. Archive names have the same form as file names.**

  Program terminates.

- **Improper syntax for a file name.**

  Program terminates.

- **Improper date syntax. The correct syntax is month/day/year (mm/dd/yr or dd-mm-yy, where yr is the short year form, e.g. "82", not "1982")**

  Program terminates.

- **Temporary restore file not found, must redo from start.**

  Program terminates.

- **Inserted diskette is not newly formatted.**

  Prompt repeats.

- **Diskette is not a completed archive diskette.**

  Prompt repeats.

- **Inserted diskette is not [diskette <N>] of Archive Library <XXXXXXX>.**

  Prompt repeats.

- **Inserted diskette is not an archive diskette.**

  Program terminates.

- **Diskette archived under incompatible Archive version <X>. You must restore under version <X> or higher.**

  Program terminates.

- **<filename> already exists on volume <X>. Restore anyway? (y/n)**

- **Drive <X> is not a double sided drive as required by your Archive Disk Library. Redo with a double sided floppy drive.**

  Program terminates.

- **Diskette full. Can not write filename.**

  Reformat this diskette using FORMAT. Also check that you are using double-sided diskettes if you specified double in your command line.

- **System error: Segment wrap around on system read. Program exit is necessary. Suggested recovery : split your current Archive Library into two sections and archive each section separately.**

  Program terminates.

- Incomplete command line. Expected to find <key word>.

  Program terminates.

- Archive Library mis-match!! The archive Library last being restored to volume <X> does not match the Archive Library of the inserted diskette.

- Error: File list does not exist on requested volume.

SECTION V

SOLVING PROBLEMS

21. SOLVING PROBLEMS

This Section describes how to locate and solve problems you have with your computer. The first chapter describes how to solve general problems and how to prepare for a service call. The second chapter describes how to respond to error messages that the computer may display.

When you have a problem, first examine the situation:

- Don't become alarmed. You can solve most problems yourself in a few minutes.

- Note if an error message is displayed and whether the computer is making unusual noises.

- Note what you were doing when the problem occurred. This includes the type of operation and the program name (and version number, if possible).

Most problems you can have while doing the procedures described in this guide are indicated by error messages. Chapter 22 lists the error messages and describes recovery procedures.

If you are using an application program when the problem occurs, refer to the program's user guide. The guide will explain how to respond to error messages displayed by the program.

If no error message is displayed, refer to "General Problem Solving."

Some computer problems must be corrected by a computer technician. If you have a problem that persists, or if your problem is not described in Table 21-1, you may need a service call. 21.3 describes how to prepare for a service call.

21.1 GENERAL PROBLEM SOLVING

Table 21-1 describes how to solve problems that are not indicated by an error message.

Table 21-1: General Problem Solving

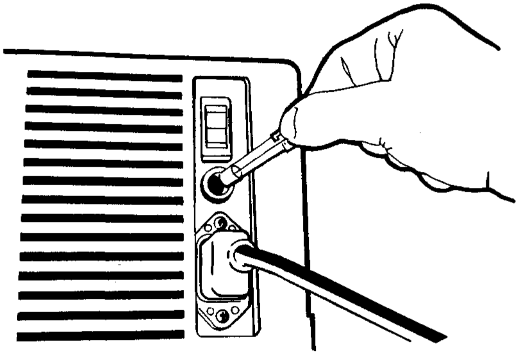
PROBLEM	SOLUTION
Display is blank and there is no fan noise.	1. Turn off the power and the power cord plug for proper connection. 2. Check the wall socket for power by plugging in another device (such as a lamp). 3. Replace the fuse (as described in 21.2).
Display is blank or very dim, but there is fan noise.	1. Turn off the power and check the display connector for proper connection. 2. Try to increase display brightness (see 2.5).
The operating system does not load and there is no drive noise.	(Call your service representative)
The operating system does not load, but there is drive noise.	1. Check that the brightness is adjusted properly. 2. Try to load the operating system with a backup system diskette.

When loading an application program, there is drive activity, but the computer "hangs up" before the first frame of the program.	Some application programs require that you first load a programming language module. Check that you loaded the proper module.
The display freezes and the computer does not respond to your commands.	1. If you are using CP/M-86, type one or more ALT-Cs. 2. Check the keyboard connector for proper connection. 3. Press the reset button.
The computer displays what you have typed, followed by a question mark.	1. Check what you have typed for errors. Retype the material correctly. 2. If you are loading a program, check that the program is on the diskette or volume you are using.
Repeated inconsistent or erratic results.	The most frequent cause of this type of problem is a worn or damaged diskette. Retry the operation or try your backup diskette.
Bell sounds.	While using the calculator field, you have typed a key that cannot be used in calculations, or you have typed a number that is more than 14 digits long.

21.2 REPLACING THE FUSE

The fuse holder is located on the rear of the processor unit (Figure 21-1).

Figure 21-1: Replacing the Fuse



If the fuse burns out, you can have your service representative change it for you or you can change it yourself. To replace the fuse:

1. Turn off the power for the main unit, and for the external hard disk if present.

2. Disconnect the power cord.

3. Open the fuse holder with a coin or screwdriver.

4. Replace the burnt out fuse with a new fuse. Only use fuses obtained from your dealer for your computer.

5. Close the fuse holder.

6. Reconnect the power cord.

7. Turn on the power.

21.3 PREPARING FOR A SERVICE CALL

Contact your service representative if you cannot solve a problem yourself. Before calling for help, make sure you have the following information:

- A complete description of the problem, including error messages.
- The program and data diskettes you were using when the problem occurred.
- Model and serial numbers of your:
  - Keyboard
  - Display unit
  - Processor unit
  - External hard disk
- Who to see if you are not in the office when help arrives. Be sure that person has your diskettes and notes on the problem.
- Your business hours and phone number.

22. MESSAGES AND ERROR CODES

The computer generally displays an error message when it encounters a problem. This chapter lists the error messages you may see while doing the procedures described in this guide. With each message is a description of how to correct the error.

Some error messages contain 2-digit error codes. The error codes are described in 22.2

22.2 ERROR MESSAGES

The error messages are listed in alphabetical order, except for the messages you may see while loading the operating system (DOS). These messages are at the front of the list.

X Ø1

DOS loading error: Check that you have inserted your system diskette label-side up and label-edge last. Reinsert the diskette correctly.

X Ø2

DOS loading error: Check that both drive doors are open. Reinsert your system diskette in drive A.

X 99

DOS loading error: The diskette in drive A does not contain an operating system. Remove the diskette and insert a system diskette.

X nn
(where nn is a number other than Ø1, Ø2, or 99)

DOS loading error: Your system diskette may be faulty. Try again with a backup diskette.

M X nnnnnn

DOS loading error: The diskette in drive A may be faulty. Try again with another diskette.

ABORTED: filespecifier

PIP error: The user has aborted a PIP operation by pressing a key.

Allocation error for file filename

CHKDSK error: The named file had a data block allocated to it that does not exist (that is, a data block number larger than the largest possible block number). CHKDSK truncates the file short of the bad block.

Archive Library mis-match!! The Archive Library last being restored to volume <X> does not match the Archive Library of the inserted diskette.

ARCHIVE error: Restart the restore using the correct Archive Library or redo from the start.

BAD BOOTCOPY COMMAND

BOOTCOPY error: The BOOTCOPY command was incorrectly typed. Re-enter the command.

Bad command or file name

MS-DOS error: You typed an incorrect command or file name. Retype the command or file name correctly.

Bad DCOPY command

DCOPY error: The command was typed incorrectly. Retype the command.

BAD Directory on d: Space Allocation Conflict:
User n d:filename.typ

STAT has detected a space allocation conflict in which one data block is assigned to more than one file. One or more file names might be listed. Each of the files listed contains a data block already allocated to another file on the disk. You may be able to correct the problem by PIPing the files to a backup disk, and then erasing the files listed. After erasing the conflicting file or files, press ALT-C to regenerate the allocation vector. NOTE: the files may contain faulty data and should be checked carefully.

Bad Drive Number: press any key to continue.

DCOPY or FORMAT error: An incorrect drive name was typed.

Bad FORMAT command

FORMAT error: The command was typed incorrectly. Retype the command.

BAD MEDIA. PRESS RETURN KEY TO START OVER OR ALT-C TO EXIT.

SDCOPY error: Utility encountered a bad sector while reading or writing a diskette. User should reformat or replace diskette.

Bdos Err on X: R/O

CP/M-86 error: Type an ALT-C to terminate the program you were using. At the OS command prompt, type another ALT-C to clear the disk system

If your diskette is write-protected, retry with a diskette that is not write-protected. If the error persists, try again with another diskette.

The drive has been assigned read-only status with a STAT command, or the disk in the drive has been changed without resetting the disk system with an ALT-C. CP/M-86 terminates the current program as soon as you press any key.

Bdos err on X: bad sector

This can indicate a hardware problem, or a worn, improperly formatted or missing disk. Enter ALT-C to terminate the program and return to CP/M-86; then enter ALT-C again to reset the disk system. Or press the Return key to ignore the error.

Cannot COPY double-sided diskettes

DCOPY error: The computer being used does not have double-sided disk drives. Try again with a single-sided diskette.

Cannot FORMAT double-sided diskettes

FORMAT error: The computer being used does not have double-sided disk drives. Try again with a single-sided diskette.

CANNOT FORMAT TRACK

FORMAT error: You possibly have a bad diskette. Run FORMAT again. If the error persists, discard the diskette.

CANNOT OPEN FILE

BOOTCOPY error: BOOTCOPY could not open the input file. Make sure that the file exists and that you spelled the filename correctly.

CANNOT READ: devicename or filename

PIP error: The source specified in the PIP command is illegal. An output device has probably been specified as a source.

CANNOT READ TRACK

DCOPY error: You possibly have a bad diskette. Run the program again. If the error persists, discard the diskette.

CANNOT SIZE TRACK

DCOPY or FORMAT error: You possibly have a bad diskette. Run the program again. If the error persists, discard the diskette.

CANNOT WRITE: devicename

PIP error: The destination specified in the PIP command is illegal. An input device has probably been specified as a destination.

CLOSE DRIVE DOOR

SDCOPY error: Drive door open during attempted read or write of sector.

COMMANDNAME?

CP/M-86 error: If CP/M-86 cannot find the command you specified, it returns the command name you entered followed by a question mark. Check that you have typed the command name correctly, or that the command you requested exists as a .CMD file on the default or specified disk.

Copy aborted at track nn

DCOPY error: The diskette may have a bad track, and the program is unable to continue. Run the program again, or try moving to another drive. If the error persists, discard the diskette.

DESTINATION IS R/O, DELETE (Y/N)?

PIP error: The destination file specified in a PIP command already exists and it is read/only. If you type Y, the destination file is deleted before the file copy is done.

Directory error-file: filename

CHKDSK error: No valid data blocks are allocated to the named file. CHKDSK deletes the file.

Disk not initialized

CHKDSK error: No directory or file allocation table was found. If files exist on the disk, and the disk has been physically harmed, it may still be possible to transfer files from this disk to recover data.

DISK READ ERROR: filespecifier

PIP error: The input disk file specified in a PIP command could not be read properly. This is usually the result of an unexpected end of file. Correct the problem in your file.

DISK WRITE ERROR: filespecifier

PIP error: A disk write operation could not be successfully performed during a PIP command, probably due to a full disk. You should either erase some unnecessary files or get another disk with more space and execute PIP again.

Diskette archived under incompatible Archive version <X>. You must restore under version <X> or higher.

ARCHIVE error: Program terminates. Restart the restore using the correct RESTORE version.

Diskette full. Can not write filename. Reformat this diskette using FORMAT not DELETE *.* and CHKDSK. Also check that you are using double-sided diskettes if you specified double in your command line.

ARCHIVE error: Reformat this diskette using FORMAT.

Diskette is not a completed archive diskette.

ARCHIVE error: The ARCHIVE prompt is repeated. Insert a completed archive diskette.

Divide by Zero

Calculator error: You have tried to divide a number by zero. Type two shifted Calcs to abandon the calculation.

DOUBLE-SIDED DISK CANNOT BE COPIED ON SINGLE-SIDED DRIVE.

SDCOPY error: Response to user's attempt to copy double-sided diskette on a single-sided disk drive.

DOUBLE-SIDED SOURCE DISK CANNOT BE COPIED TO SINGLE-SIDED DESTINATION.

SDCOPY error: Response to user's attempt to copy double-sided diskette to single-sided diskette. Utility will again prompt user to insert a destination diskette. User should replace destination diskette with a double-sided one.

Drive not available for CP/M reading

RDCPM error: You gave an invalid drive specifier with the CP/M file name. Retype the command with a valid drive specifier.

Drive <X> is not a double-sided drive as required by your Archive Disk Library. Redo with a double-sided floppy drive.

ARCHIVE error: Program terminates. Restart using a double-sided diskette.

DRIVE DOOR OPENED

DCOPY or FORMAT error: Close the drive door and restart the program.

Drive=nn, Track=nn, Sector=nn: Error=nn
Bdos Err on X: Bad Sector

CP/M-86 error: This message indicates a hardware problem or a worn, improperly formatted, or missing diskette. Error code numbers are described in Table 22-1.

Press Return. If the message appears again, type an ALT-C to terminate the program and return to CP/M-86; then type another ALT-C. If the error persists, try again with another diskette.

ERROR: BAD PARAMETER

PIP. An illegal parameter has been entered in a PIP command. Retype the entry correctly.

ERROR: CLOSE FILE filespecifier

PIP. An output file cannot be closed. Take appropriate action after checking to see if the correct disk is in the drive and that the disk is not write-protected.

ERROR: DISK READ filespecifier

PIP. The input disk file specified in a PIP command could not be read properly. This is usually the result of an unexpected end of file. Correct the problem in your file.

ERROR: DISK WRITE filespecifier

PIP. A disk write operation could not be successfully performed due to a full disk. You should either erase some unnecessary files or get another disk with more space and execute PIP again.

Error: File list does not exist on requested volume.

ARCHIVE error: Check that your file list file is on the volume specified in your command line.

ERROR: FILE NOT FOUND filespecifier

PIP. Input file specified does not exist.

ERROR: HEX RECORD CHECKSUM filespecifier

PIP. A hex record checksum was encountered during the transfer of a hex file. The hex file with the checksum error should be corrected, probably by recreating the hex file.

ERROR: INVALID DESTINATION

PIP. The destination specified in your PIP command is illegal. You have probably specified an input device as a destination.

ERROR: INVALID FORMAT

PIP. The format of your PIP command is illegal. See the description of the PIP command.

ERROR: INVALID HEX DIGIT filespecifier

PIP. An invalid hex digit has been encountered while reading a hex file. The hex file with the invalid hex digit should be corrected, probably by recreating the hex file.

ERROR: INVALID SEPARATOR

PIP. You have used an invalid character for a separator between two input filenames.

ERROR: INVALID SOURCE

PIP. The source specified in your PIP command is illegal. You have probably specified an output device as a source.

ERROR: INVALID USER NUMBER

PIP. You have specified a User Number greater than 15. User Numbers are in the range Ø-15.

ERROR: NO DIRECTORY SPACE filespecifier

PIP. There is not enough directory space for the output file. You should either erase some unnecessary files or get another disk with more directory space and execute PIP again.

ERROR: QUIT NOT FOUND

PIP. The string argument to a Q parameter was not found in your input file. See if you have upper case/lower case conflicts.

Error reading CP/M disk

RDCPM error: A disk error occurred while trying to read the CP/M file. Try to run the program again.

Error reading CP/M disk label

RDCPM error: The disk label on the CP/M diskette cannot be read. The CP/M diskette is probably unformatted or bad.

ERROR: START NOT FOUND

PIP. The string argument to an S parameter could not be found in the source file. Check to see whether you have upper/lower case conflicts.

ERROR: UNEXPECTED END OF HEX FILE filespecifier

PIP. An end of file was encountered prior to a termination hex record. The hex file without a termination record should be corrected, probably by recreating the hex file.

ERROR: USER ABORTED filespecifier

PIP. The user has aborted a PIP operation by pressing a key.

ERROR: VERIFY filespecifier

PIP. When copying with the V option, PIP found a difference when rereading the data just written and comparing it to the data in its memory buffer. Usually this indicates a bad copy on the destination disk or drive.

ERROR WHILE READING ON DRIVE drivename
Abort,Ignore,Retry:

MS-DOS error: Type A (for abort), I (for ignore), or R (for retry). Retry the operation before ignoring or terminating (aborting) the program. Ignoring the error could result in bad data. Check the resulting file carefully.

ERROR WHILE WRITING ON DRIVE drivename
Abort,Ignore,Retry:

MS-DOS error: Type A (for abort), I (for ignore), or R (for retry). Retry the operation before ignoring or terminating (aborting) the program. Ignoring the error could result in bad data. Check the resulting file carefully.

FILE ALLOCATION TABLE BAD FOR DRIVE drivename

MS-DOS error: Check that the diskette is formatted. (The message indicates that the copy in memory of one of the allocation tables has pointers to nonexistent blocks.)

FILE EXISTS

CP/M-86 error: CP/M-86 has been asked to create a new file using a file specification that is already assigned to another file. Either delete the existing file or use another file specifier.

File name not found

ARCHIVE error: Program continues missing filename is reported.

File not found

MS-DOS or CP/M-86 error: The file you specified does not exist. Check that you specified the file correctly and try again.

FILE READ ERROR

BOOTCOPY error: BOOTCOPY could not read the input file. The file is probably corrupt. Try to re-create the file.

File size error for file filename

CHKDSK error: The size of the file in its directory differs from its actual size. The size in the directory is automatically adjusted to indicate its actual size on the disk. (The amount of useful data may be less than the size shown because the last data block may not be used fully.)

**<filename> already exists on volume <X>.
Restore anyway? (y/n)**

ARCHIVE error: Type Y to continue the restore or N to skip the file and go to the next file in the restore library.

Files cross-linked: filename and filename

CHKDSK error: The same data block is allocated to both files. CHKDSK does not try to correct this. To correct the problem, first use the COPY command to make copies of both files; then delete the originals. Review each file for validity and edit as necessary.

Format aborted at track nn

FORMAT error: The diskette has a bad track, and the program is unable to continue. Run the program again, or try moving to another drive. If the error persists, discard the diskette.

Format failure

FORMAT error: The format was unsuccessful. This message occurs only with one of the other FORMAT error messages.

Hard disk full; <filename> cannot be written.

ARCHIVE error: Start over using a hard disk volume with more space. Program terminates.

Improper date syntax. The correct syntax is month/day/year. (mm/dd/yr where yr is the short year form, e.g. "82", not "1982")

ARCHIVE error: Program terminates. Try again using correct syntax. The American format is mm/dd/yy. The European format is dd-mm-yy.

Improper response.

ARCHIVE error: The ARCHIVE prompt is repeated. Respond to the prompt correctly.

Improper syntax for an archive name. Archive names have the same form as file names.

ARCHIVE error: Program terminates. Try again using correct syntax.

Improper syntax for a file name.

ARCHIVE error: Program terminates. Try again using correct syntax.

Incomplete command line. Expected to find <key word>.

ARCHIVE error: Program terminates. Retype the command line correctly.

Inserted diskette is not an archive diskette.

ARCHIVE error: Program terminates. You must restart the archive.

Inserted diskette is not [diskette <N>] of Archive Library <XXXXXX>.

ARCHIVE error: The ARCHIVE prompt is repeated. Insert the correct diskette.

Inserted diskette is not newly formatted.

ARCHIVE error: The ARCHIVE prompt is repeated. Insert a newly formatted diskette.

Insufficient disk space

RDCPM error: There is not enough room on the MS-DOS diskette to transfer the CP/M file. Use a different MS-DOS diskette with more available space.

Invalid Assignment

STAT. An invalid device was specified in a STAT device assignment. Use the STAT VAL: display to list the valid assignments for each of the four logical STAT devices: CON:, AXI:, AXO:, and LST:.

Invalid CP/M disk type

RDCPM error: The diskette that is being read from is not a CP/M diskette. Try again with a CP/M diskette.

Invalid date

MS-DOS error: You typed the date incorrectly. Retype the date correctly. Use numbers only; separate the numbers with hyphens (-) for European format (day-month-year) or slashes (/) for American format (month/day/year).

Invalid Disk Assignment

STAT error: An invalid STAT drive command was given. The only valid drive assignment in STAT is STAT d:=R/O.

Invalid drive specification

FORMAT error: An invalid drive was specified with the command. Retype the command with the correct drive specifier.

Invalid drive specification

MS-DOS error: You specified an invalid drive. Retype the command.

Invalid File Indicator

STAT error: This message results from an invalid command to set file attributes. These are the only options valid in a STAT filespecifier [option] command: ro, rw, sys, dir.

INVALID FORMAT: invalidentry

PIP error: The format of your PIP command is illegal. See the description of the PIP command.

Invalid parameter

FORMAT error: An invalid command switch was specified with the FORMAT command. Retype the command with valid switches.

Invalid time

MS-DOS error: You typed the time incorrectly. Retype the time correctly. Separate the numbers with colons (:). Do not type the time for the half second.

nnnnnn bytes of disk space freed

CHKDSK error: Disk space shown as allocated was not actually allocated and has been freed.

NO BOOT SECTORS ON SOURCE DISK

BOOTCOPY error: The source disk does not have boot tracks. Therefore there is no operating system to copy. Use a source diskette with boot tracks.

NO DISKETTE OR BAD TRACK

FORMAT error: If you have not inserted a diskette in the drive to be formatted, do so and restart the program. If the drive contains a diskette, take it out and retry with another diskette.

Use: [rw] [ro] [sys] or [dir]

STAT. This message results from an invalid set file attributes command. These are the only options valid in a STAT filespec [option] command.

Use: STAT d:=RO

STAT. An invalid STAT drive command was given. The only valid drive assignment in STAT is STAT d:=RO.

NO FILE: = filespecifier

PIP error: Input file specified does not exist.

No room in directory to create files

RDCPM error: The directory on the MS-DOS diskette is full. Use a different MS-DOS diskette with fewer files.

NOT A SYSTEM IMAGE

BOOTCOPY error: The boot tracks on the source disk do not contain an operating system. Use a system disk.

NOT ENOUGH MEMORY

BOOTCOPY error: There is not enough memory in the system to do the BOOTCOPY. Boot up a smaller operating system and then run BOOTCOPY or try to build a smaller system.

Open <filename>, cannot be closed.

ARCHIVE error: Check that current diskette is correct. Press Return when ready (Q for emergency exit).

Operating system version mismatch

FORMAT error: The operating system being used is not compatible with the version of FORMAT being used. Use the correct version of the program or load the correct operating system.

Overflow

Calculator error: Your calculation is longer than 14 digits. Type two shifted Calcs to abandon the calculation.

QUIT NOT FOUND: = source

PIP error: The string argument to a Q parameter was not found in your input file. Check for upper/lower case conflicts.

Read error: D=nn, T=nn, S=nn, E=nn
CANNOT READ TRACK

DCOPY or FORMAT error: Retry the program; discard the diskette if the error persists. (D stands for drive, T stands for track, S stands for sector, and E stands for error code. Error codes are described in Table 22-1.)

Sector write error: D=nn, T=nn, S=nn, E=nn
CANNOT WRITE DISK LABEL

DCOPY or FORMAT error: The program is unable to write the disk label to the diskette. The drive, track, sector, and error code numbers are indicated. Error codes are described in Table 22-1.

Soft format error: D=nn, T=nn, S=nn, E=nn

FORMAT error: A soft format error has occurred. The drive, track, sector, and error code numbers are indicated. Error codes are described in Table 22-1.

If there are more than 9 soft errors on a track, the program aborts. Soft errors in moderate numbers are not harmful to diskette performance.

Source and destination drives must not be the same

RDCPM error: The CP/M file cannot be moved onto the same diskette in MS-DOS format. Specify separate drives in the command.

Source file name missing

RDCPM error: You typed the RDCPM command without a CP/M file specifier. Retype the command with a CP/M file specifier.

Source file not found

RDCPM error: The CP/M file name given cannot be found on the CP/M diskette. Retype the command with the correct file name or insert the correct CP/M diskette.

START NOT FOUND: = source

PIP error: The string argument to an S parameter could not be found in the source file. Check for upper/lower case conflicts.

System error: Segment wrap around on system read. Program exit is necessary. Suggested recovery : split your current Archive Library into two sections and archive each section separately.

ARCHIVE error: Program terminates. Recover as suggested.

Temporary restore file not found, must redo from start.

ARCHIVE error: Program terminates. Redo the restore.

Too Many Files

STAT. A STAT wildcard command matched more files in the directory than STAT can sort. STAT can sort a maximum of 512 files.

VERIFY ERROR: filespecifier

PIP error: When copying with the V option, PIP found a difference when rereading the data just written and comparing it to the data in its memory buffer. Usually this indicates a bad copy on the destination disk or drive.

VOLUME IN DRIVE IS NONEXISTENT OR IS NOT A FLOPPY.

SDCOPY error: Either user is attempting to use a volume on a hard disk drive, or there is no floppy in the disk drive.

Volume <X> is available but is empty.

ARCHIVE error: Program terminates. Try again with the correct volume.

Volume <X> not available.

ARCHIVE error: Program terminates. Try again with the correct volume.

WARNING - DOUBLE-SIDED DESTINATION DISK WILL BECOME SINGLE-SIDED AS A RESULT OF THIS OPERATION. DO YOU WISH TO CONTINUE? (y/n)

SDCOPY error: Response to user's attempt to copy single-sided diskette to double-sided diskette. Type "y" for yes or "n" for no.

WRITE ERROR: Drive=xx, Track=xx, Sector=xx, Error=xx

BOOTCOPY error: There was a write error on the given track and sector. Try to run the program again to see if the same error occurs.

WRONG DESTINATION DISK LABEL TYPE

BOOTCOPY error: The destination disk has an incorrect software label. Format the disk with the correct version of FORMAT; any files on the disk will be lost.

WRITE PROTECTED DISKETTE

FORMAT error: The diskette being formatted has a write-protect label on it. Remove the label and try again. Files already existing on the diskette may be erased if you do this.

WRONG DESTINATION DISK

SDCOPY error: Current destination diskette is different from original one. Applies to situation involving multiple insertions of destination diskette.

WRONG SOURCE DISK

SDCOPY error: Current source diskette is different from original one. Applies to situation involving multiple insertions of source diskette.

<X> is not an available floppy drive.

ARCHIVE error: Program terminates. Try again with an available floppy drive.

22.3 ERROR CODES

Several MS-DOS and CP/M-86 error messages contain 2-digit error codes. There are five general types of error codes:

1x-- Cannot find sector

2x-- Cannot read sector

3x-- Sector verify error

4x-- Formatting error

Fx-- Miscellaneous

Table 22-1 defines the error codes.

Table 22-1: Error Codes

ERROR CODE NUMBER	DEFINITION
	COULD NOT COMPLETE READ-RELATED COMMAND
11	Noise encountered on sync line
12	Bad header block ID
13	Checksum error in header
14	Header GCR error
15	Wrong track
16	Wrong sector
17	Bad job code
	INVALID DATA ON DISKETTE
21	Bad data block ID
22	Checksum error in data
23	GCR error
24	Sync time out
	DEFECTIVE DRIVE OR DISKETTE
31	Bad data block ID
32	Verify error
33	Checksum error
34	GCR error
	FORMAT PROGRAM ERROR CODES
41	No sync found (bad missing diskette)
42	Bad header ID
43	Wrong track
44	Wrong sector

45	Bad header checksum
46	Gap error
47	GCR error
48	No data sync
49	Bad data ID
4A	Data verify error
4B	Data checksum
4C	Gap 2 error
4D	GCR error

MISCELLANEOUS

F1	Cannot address second side of diskette
F2	Step error - cannot find track
F3	Data not written due to disk change
F4	Cannot write to disk until logged (^C)
F5	Wrong diskette type
F6	Cannot start disk operation
F7	Illegal track number
F8	Illegal drive number
F9	Illegal disk operation
FA	Door open
FB	Drive motor not up to speed
FC	Write-protected diskette
FD	Bad track on diskette
FE	Cannot complete disk operation
FF	Bad diskette or unformatted diskette

SECTION VI

APPENDIXES

APPENDIX A: DISPLAY DRIVER SPECIFICATIONS

A.1 INTRODUCTION

The software in the BIOS display interface receives ASCII characters and either displays them or uses them for display control. Characters in the ASCII control set (hex ØØ through hex 1F and hex 7F) are control characters and are not displayed (see the ESC 8 exception). These characters may, however, affect the display. Most of the control characters act independently. But for some actions, more than one control character is required to specify the action. This is done by using the ASCII escape code (hex 1B) followed by one or more characters. The control characters and the escape sequences are described in A.2 and A.3.

The display controls are accessed from high-level languages by using the PRINT or equivalent statements. For example, in MS-BASIC —

PRINT CHR$(27)+CHR$(112)

turns on reverse video.

NOTE: Decimal 27 equals hex 1B, which is Escape and decimal 112 is hex 7Ø.

A.2 CONTROL CHARACTERS

o Bell (ALT-G, hex Ø7): This is not really a display control character. It sends the CODEC a series of signals that cause it to make the sound of a bell.

- Backspace (ALT-H, hex Ø8): Moves the cursor one column to the left. If cursor is at column 1, it moves to column 8Ø of the previous row (unless cursor is at column 1, row 1, in which case it moves to column 8Ø, row 1).

- Horizontal Tab (ALT-I, hex Ø9): Moves the cursor to the next tab stop. Tab stops are fixed at columns 9, 17, 25, 33, 41, 49, 57, 65, and 72 through 8Ø. If the cursor is at column 8Ø, it remains there.

- Line Feed (ALT-J, hex ØA): Moves the cursor down one row. If cursor is at row 24, then the display scrolls up one row. (The Line feed can also be treated as a Return. See ESC x9.)

- Carriage Return (ALT-M, hex ØD): Moves the cursor to column 1 of the current row. (Return can also be treated as a line feed. See ESC x8.)

- Shift In (ALT-N, hex ØE): Shifts to character set 1 (G1).

- Shift Out (ALT-Ø,hex ØF): Shifts to character set Ø (GØ).

A.3 ESCAPE SEQUENCES

Table A-1: Cursor Functions

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
Esc-A	1B, 41	Moves the cursor up one line.
Esc-B	1B, 42	Moves the cursor down one line without changing columns.
Esc-C	1B, 43	Moves the cursor one character position to the right.
Esc-D	1B, 44	Moves the cursor one character position to the left.
Esc-H	1B, 48	Moves the cursor to the home postion (upper left corner of screen).
Esc-I	1B, 49	Moves the cursor up one line, and stays in the same column.
Esc-n	1B, 6E	Reports the cursor position.
Esc-j	1B, 6A	The display driver saves the cursor position.
Esc-k	1B, 6B	Returns the cursor to the previously saved cursor position.

Table A-1 (Continued)

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
Esc-Y[l][c]	1B, 59	Moves the cursor via direct cursor addressing, where l is the line number (hex) and c is the column number (hex). The first line and the left column are both 2Ø hex (the smallest value of the printing characters) and increase from there. Since the lines are numbered from 1 to 19 hex (from top to bottom) and the columns from 1 to 5Ø hex (from left to right), you must add the proper line and column numbers to 1F. No movement occurs if l and/or c are invalid.

Table A-2: Editing Functions

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
Esc-E	1B, 45	Erases the entire screen.
Esc-b	1B, 62	Erases from the start of the screen up to (and including) the cursor position.
Esc-J	1B, 4A	Erases from the cursor position to the end of the page.
Esc-l	1B, 6C	Erases entire line.
Esc-o	1B, 6F	Erases from the beginning of line up to (and including) the cursor position.
Esc-K	1B, 4B	Erases from cursor position to the end of the line.
Esc-L	1B, 4C	Inserts a blank line. The current line and all following lines scroll down one line. The cursor moves to the beginning of the blank line.
Esc-M	1B, 4D	Moves cursor to beginning of line, deletes the line, and then scrolls all following lines up one line. A blank line is inserted at line 24.

Table A-2 (Continued)

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
Esc-N	1B, 4E	Deletes character at cursor position and shifts the rest of the line one character position to the left.
Esc-@	1B, 4Ø	Enters the Insert Character mode. This lets you insert characters into screen text. As each new character is inserted, the character at the end of the line is lost.
Esc-O	1B, 4F	Exits Insert Character mode.

Table A-3: Configuration Functions

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION	
Esc-x[Ps]	1B, 78	Sets mode(s) as follows:	
		Ps	Mode
		1	Enable 25th line.
		3	Hold screen mode on.
		4	Use block cursor.
		5	Cursor off.
		8	Automatic line feed after a Return.
		9	Automatic Return after a line feed.
		A	Increase audio volume.
		B	Increase CRT brightness.
		C	Increase CRT contrast.
Esc-y[Ps]	1B, 79	Resets mode(s) as follows:	
		Ps	Mode
		1	Disable 25th line.
		3	Hold screen mode off.
		4	Use underscore cursor.
		5	Cursor on.
		8	No auto line feed.
		9	No auto Return.
		A	Decrease audio volume.
		B	Decrease CRT brightness.
		C	Decrease CRT contrast.

Table A-3 (Continued)

	ASCII CODE	
ESCAPE	GENERATED	
SEQUENCE	(HEXADECIMAL)	SEQUENCE DEFINITION
Esc-^	1B, 5E	Toggle hold mode.
Esc-[	1B, 5B	Set hold mode.
Esc-\	1B, 5C	Clear hold mode.
Esc-|	1B, 7C	Activate user-defined console (T.B.A.).

Table A-4: Operation Mode Functions

	ASCII CODE	
ESCAPE	GENERATED	
SEQUENCE	(HEXADECIMAL)	SEQUENCE DEFINITION
Esc-p	1B, 7Ø	Enters reverse video mode.
Esc-q	1B, 71	Exits reverse video mode.

Table A-5: Special Functions

	ASCII CODE	
ESCAPE	GENERATED	
SEQUENCE	(HEXADECIMAL)	SEQUENCE DEFINITION
Esc-}	1B, 7D	Disables the keyboard.
Esc-{	1B, 7B	Enables the keyboard.
Esc-v	1B, 76	Enables wrap-around at the end of the line.
Esc-w	1B, 77	Disables wrap-around at the end of the line.

Table A-5 (Continued)

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
Esc-z	1B, 7A	Resets terminal to power-on configuration.
Esc-$	1B, 24	Transmits the character at cursor location.
Esc-]	1B, 5D	Transmits the 25th line.
Esc-#	1B, 23	Transmits the page.
Esc-(	1B, 28	Sets high intensity.
Esc-)	1B, 29	Sets low intensity.
Esc-+	1B, 2B	Clears the foreground. (High-intensity displayed characters.)
Esc-Z	1B, 5A	Identifies display as emulating VT52 (the terminal responds with an Esc-K).
Esc-Ø	1B, 3Ø	Sets the underline mode.
Esc-1	1B, 31	Resets the underline mode.
Esc-2	1B, 32	Enables cursor blink.
Esc-3	1B, 33	Disables cursor blink.
Esc-8	1B, 38	Sets the test (literally) mode for the next single character.

Table A-5 (Continued)

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
Esc-i[n]	1B, 69	Displays the system sign-on banner, as follows (n is the ASCII numeric character): n Display Ø Entire banner. 1 Company logo only. 2 Product name only. 3 Configuration information only.

A.4 132-COLUMN UTILITY

DESCRIPTION

The 132-column utility (132C) provides a simulated 132-column display in the 8ØØ-dot by 4ØØ-line HIRES display mode of your computer. The characters are displayed in a 5-by-7 dot matrix in a 6-by-1Ø cell to give the 132-column display. In addition, a standard display of 8Ø columns by 25 lines is simulated with an 8-by-11 dot matrix in a 1Ø-by-16 cell. The 132 column mode is enabled when the normal BIOS display interface recognizes the appropriate escape sequence.

INSTALLATION

Enter this sequence to activate the 132 Column feature for use with an application program:

```
A>132C(cr)
A>132C-ON(cr)
```

These commands can be combined in a batch (.BAT) file, if you like.

The 132C feature is turned off with the command:

```
A>132C-OFF <cr>
```

The 132C program installs itself into the BIOS. The program requires about 50K of memory: the 800-by-400-line HIRES screen requires 40K; the character sets and code use the rest. The 132C program copies itself into the lowest 64K block of memory and removes its 50K required bytes from the system until you press the reset button or turn off your computer.

You can access 132C, if installed from an application program, by sending an ESC | (1B 7C) to the display interface. This is done from MS-BASIC with:

```
PRINT CHR$(27)+CHR$(124)
```

From MS-BASIC, 132C is turned off with an ESC z (1B 7A):

```
PRINT CHR$(27)+"z"
```

The display control sequences described above are also available in 132C mode.

When you've finished and saved your work, press the reset button to release the memory allocated to 132C and return to normal character mode.

APPENDIX B: CHARACTER SETS

B-1: International Character Set

HIGH NIBBLE

LOW NIBBLE ▶	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0:		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1:	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2:		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4:	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5:	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7:	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	⌂
8:	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9:	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
A:	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B:	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C:	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D:	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E:	α	β	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F:	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

B-2: British Standard Character Set

HIGH NIBBLE

LOW NIBBLE ➧	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0:		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1:	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2:		!	"	£	$	%	&	'	(	)	*	+	,	-	.	/
3:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4:	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5:	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7:	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	⌂
8:	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9:	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
A:	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B:	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C:	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D:	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	¢
E:	α	β	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F:	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

B-3: French Standard Character Set

LOW NIBBLE ➧

HIGH NIBBLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0:		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1:	►	◄	↕	‼	¶	@	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2:		!	"	£	$	%	&	'	(	)	*	+	,	-	.	/
3:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4:	à	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5:	P	Q	R	S	T	U	V	W	X	Y	Z	°	ç	§	^	_
6:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7:	p	q	r	s	t	u	v	w	x	y	z	é	ù	è	¨	⌂
8:	~	ü	{	}	â	ä	å	ê	ë	[	]	ï	î	ì	Ä	Å
9:	É	æ	Æ	ô	ö	ò	û	ÿ	Ö	Ü	\	¢	£	¥	₧	ƒ
A:	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B:	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C:	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D:	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	¢
E:	α	β	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F:	≡	±	≥	≤	⌠	⌡	÷	≈	!	∙	·	√	ⁿ	²	■	

B-4: German Standard Character Set

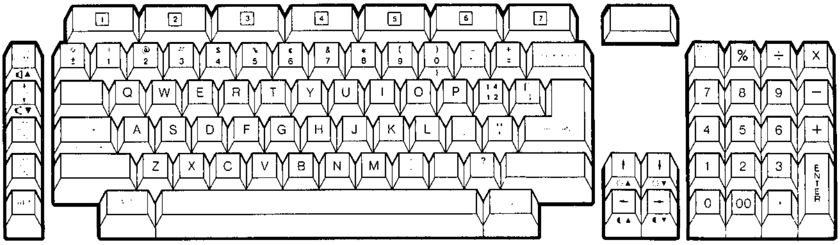
LOW NIBBLE ➧

HIGH NIBBLE

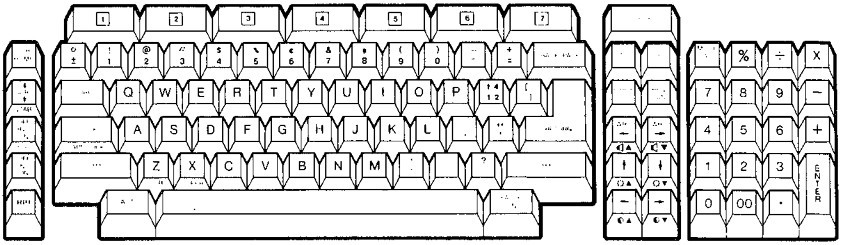
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0:		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1:	►	◄	↕	‼	¶	@	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2:		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4:	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5:	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ü	^	_
6:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7:	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ü	ß	⌂
8:	~	{	}	â	à	å	ê	è	é	ë	[	]	î	ì	ï	Å
9:	É	æ	Æ	ô	ò	ù	û	ÿ	ç	°	\	¢	£	¥	R	ƒ
A:	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B:	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C:	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D:	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	¢
E:	α	Γ	λ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F:	≡	±	≥	≤	⌠	⌡	÷	≈	¦	∙	·	√	ⁿ	²	■	

APPENDIX C: KEYBOARDS

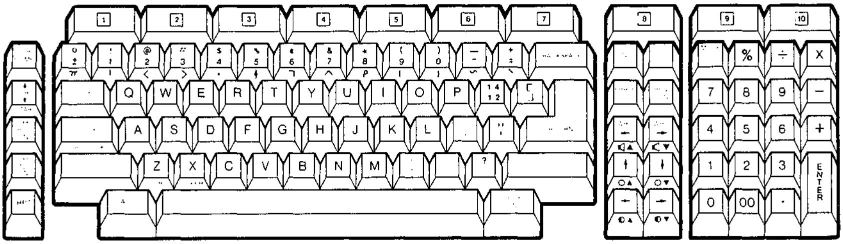
C-1: Standard Keyboard



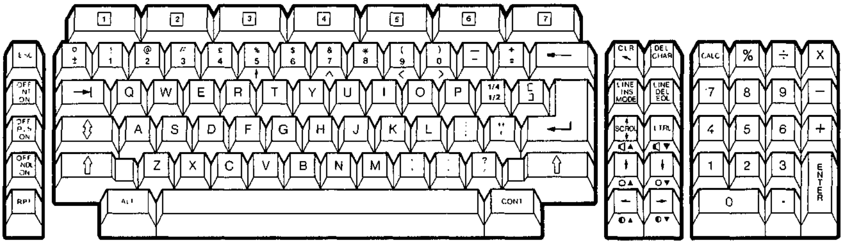
C-2: Word Processing Keyboard



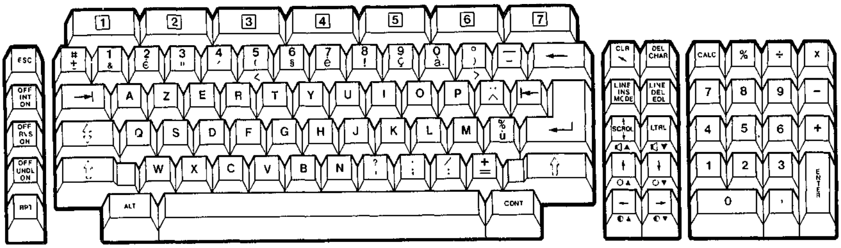
C-3: Programming Keyboard



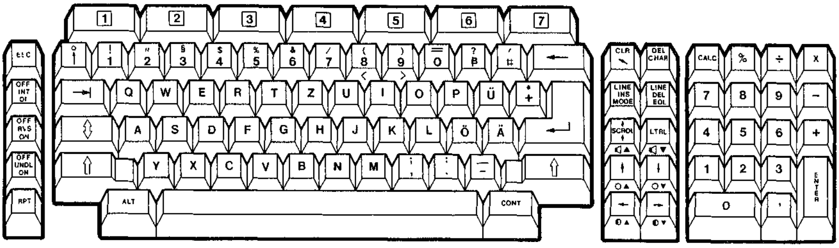
C-4: British Keyboard



C-5: French Keyboard



C-6: German Keyboard



APPENDIX D: SOFTWARE DEVELOPMENT TOOLS

This appendix gives you an idea of the range of Software Development Tools available for use with your computer. Contact your dealer for details on these and other options.

D.1 PROGRAMMER'S TOOL KIT

The **Programmer's Tool Kit** is a package of utilities designed for applications and systems programmers. All of the utilities run under the MS-DOS operating system.

D.1.1 VOLUME I

- FABS/86: A module that uses key files for fast data retrieval. FABS/86 organizes the key files in balanced trees (Btrees), eliminating the need to search the entire data file each time you want to retrieve a piece of information.

- AUTOSORT/86: A sort/merge/select utility that helps you sort very large files that contain fixed-length fields within fixed-length records. AUTOSORT/86 can be used by itself or it can be called as a subroutine from application programs.

(NOTE: Royalties on both FABS/86 and AUTOSORT/86 have been paid by VICTOR Technologies, Inc. Therefore, these products may be included in any software developed for distribution on our products without further payment of royalties.)

- PMATE-86: A full-screen, expandable editing system that combines the best features of text editors, word processors and text output processors. PMATE-86 lets you create and maintain text files, and is very useful for editing programs.

- EFONT: A font editor used to create or modify the characters that appear on the screen. With EFONT, you can make changes to an existing character set or you can create a new one to use in a special application.

- KEYGEN: A utility that lets you take advantage of your computer's "soft" keyboard. KEYGEN is used to define and change the functions performed by individual keys.

- MODCON: A utility you can use to modify the configuration of your operating system. With MODCON, you can choose the keyboard and character set tables that you want to use.

D.1.2 VOLUME II

- MS-CREF: A cross-reference utility that helps you debug assembly language programs. MS-CREF produces an alphabetical listing of all symbols used in a program, making it easy to locate the program line number where a particular symbol occurs. MS-CREF can be used with the MACRO-86 macro assembler.

- MS-LINK: A relocatable linker that links together separately produced modules of 8Ø86/8Ø88 object code. The code can be generated from MACRO-86 or any of the Microsoft compilers. MS-LINK is user-friendly. It prompts you each time a command needs to be issued; your answers to the prompts are the commands.

- MS-LIB: Creates and modifies library files used by MS-LINK. MS-LIB adds, deletes or extracts modules from a library. It can create a general library used by many programs or for the structured development of a single program.

- DEBUG: A debugging program that provides a controlled testing environment for binary and executable object files. DEBUG eliminates the need to reassemble a program to see if you can fix a problem by making a minor change.

- MACRO-86: A powerful assembler for 8086/8088-based computers. MACRO-86 has many features usually found only in large computer assemblers. Macro assembly, conditional assembly, and a variety of assembler directives give you the tools you need to get full use and power from an 8086/8088 microprocessor.

- SYSELECT: A system generation program that lets you create a custom operating system. SYSELECT lets you customize keyboard tables, character sets, default printer types, serial port specifications, logos, and banners.

D.2 GRAPHICS TOOL KIT

The **Graphics Tool Kit** is a package of utilities that help you create screen graphics for special purposes. The Tool Kit includes:

- BUSIGRAF: A business graphics package that lets you create and edit pie charts, bar graphs, line plots, and organization charts.

- GRAFIX: A powerful set of programming commands that help you create screen graphics. With GRAFIX, you can draw different types of lines, circles and arcs, fill in areas and bars (and choose the fill-in pattern), and many other functions.

- CHARGRAF: A character graphics system that lets you create and print graphics with characters instead of high-resolution graphics.

- EFONT: A font editor used to create or modify the characters that appear on the screen. With EFONT, you can make changes to an existing character set or you can create a new one to use in a special application.

- KEYGEN: A utility that lets you take advantage of your computer's "soft" keyboard. KEYGEN is used to define and change the functions performed by individual keys.

- MODCON: A utility you can use to modify the configuration of your operating system. With MODCON, you can choose the keyboard and character set tables that you want to use.

- GW BASIC: An interactive interpreter that supports music and high-resolution graphics.

D.3 COMMUNICATIONS

With these packages, your computer can communicate with and control external devices and other computers, including the IBM Personal Computer. Current communications packages include:

- IEEE 488: A parallel communications interface used for device communication and control. You can use IEEE 488 to communicate with devices such as data acquisition and control products, measurement devices, plotters, and printers. These devices must conform to ANSI Standard MC1.1 "Digital interface for programmmable instrumentation."

- o PC COMM: This package lets your computer communicate with the IBM Personal Computer (PC). PC COMM lets you transmit source and object files from the IBM PC to your computer with an effective transfer rate of about 1ØØØ bytes per second. PC COMM can be used without purchasing additional hardware for the IBM PC.

- o ASYNC: This package lets your computer communicate with other computers. ASYNC lets you log on to a remote timesharing computer or transfer files between computers over hard wires or the telephone.

- o BISYNC-86/378Ø: A communications protocol emulator that runs on your computer under the CP/M-86 operating system. BISYNC-86/378Ø allows communication using IBM 277Ø, 278Ø, 3741 or 378Ø protocol, and provides for microcomputer-to-microcomputer communications mode. You use your computer as a Remote Job Entry terminal connected to a mainframe computer by telephone.

D.4 AUDIO TOOL KIT

The audio package lets you record, edit, permanently store, and play back sounds. Using a CODEC, human voice or other sounds are recorded and recalled. You can use the audio package for voice prompts, error or help messages, voice confirmation of data entered on the keyboard, and electronic voice messaging.

SECTION VII

GLOSSARY

GLOSSARY

Allocation unit: The "building blocks" used to create files on the hard disk. An allocation unit can contain 1K, 2K, 4K, 8K, 16K, 32K, or 64K bytes. You select an allocation unit size for each hard disk volume when you set up the hard disk.

ALT-key character: The letter or number key used with the ALT key to perform an ALT-key function.

ALT-key function: A function performed by holding down the ALT key and pressing another key. The ALT-key function for some keys is shown on the front key label.

Ambiguous: Used to describe a file specifier that contains wild-card characters.

Application program: A program that performs specific tasks, such as word processing or payroll accounting. The term "application" refers to the fact that the program applies the computer to a particular task.

AU: Allocation unit.

Back up: The act of creating a backup diskette.

Backup or backup diskette: A duplicate diskette. You should routinely create backups to ensure against diskette or hard disk damage with resulting data loss.

Boot: See "load."

Boot volume: See "primary boot volume."

Calculator field: A space on the screen where the computer displays calculations as they are performed by the CALC utility.

Calculator keys: The group of keys on the right of the keyboard, arranged like a calculator.

Character: A letter, number, or symbol.

Command: What you type at the keyboard to tell the computer what to do. The program you are using must contain the command in its "vocabulary" for the command to work. Commands may consist of words, parts of words, or codes.

Command line: A command and its parameters.

Command parameter: A word or group of characters that you type with the command to change the effect of the command.

Command prompt: A program's signal that requests a command from you. The DOS command prompt is a letter followed by > (for example, A>).

Command switch: A certain type of command line parameter you can use with some utilities. Each utility has its own command switches.

CP/M-86: One of the operating systems used with the computer.

CRT: The display unit. CRT stands for "cathode ray tube."

Cursor: A rectangular marker displayed on the screen to show where your next typed entry will appear.

Data: Information.

Default drive: The floppy disk drive or hard disk volume you are now using. The default drive is shown in the DOS command prompt.

Default value: The assumed value. The computer assumes a default value unless you specify another value.

Directory: See "file directory."

Disk operating system: See "operating system."

Diskette: See "floppy diskette."

Display unit: One of the three main computer units. The display unit contains the TV-like screen. The screen displays both what you type and the computer's messages to you.

DOS: The operating system. DOS stands for "disk operating system."

Double-sided diskette: A floppy diskette that can record information on both sides.

Double-sided drive: A floppy disk drive that can be used with single-sided or double-sided diskettes.

Drive: (1) A floppy disk drive. (2) In general, a floppy disk drive or a hard disk volume. Floppy disk drives and hard disk volumes are both identified with a drive designation.

Drive designation: A one-letter/colon code that identifies a floppy disk drive or hard disk volume. A drive designation is an optional part of a file specifier.

Drive door: The latched door guarding a floppy disk drive.

Error message: A message that the computer displays to notify you of a problem.

Error code: A 2-digit code shown with some error messages.

External DOS command: A command which must be loaded from disk before execution.

File: A named collection of letters, numbers, symbols, or programming code recorded on a diskette or hard disk volume.

File directory: A list of the files on a diskette or volume. Operating systems and application programs maintain a directory for each diskette or volume. The DOS updates the directory when files are recorded or erased.

File extension: An optional part of a file specifier that contains up to 3 characters and is preceded by a period (.). An extension generally identifies the file type. Some file extensions are defined by the DOS. For example, MS-DOS uses the extension .COM for files that contain DOS commands.

File name: A name you (or another user) give to a computer file. A file name contains up to 8 characters.

Floppy diskette: A circular sheet of magnetically coated flexible plastic enclosed in a firm, square jacket. Diskettes are used for storing data.

Floppy disk drive: A motor-driven component of the processor unit used to record data on or retrieve data from diskettes.

Format: To prepare a diskette to accept data so that you can record information on it.

Function key: A key that performs a function, usually defined by the program you are using. The **general function keys** are the top row of typewriter keys. The **specific function keys** are the double-key column between the typewriter keys and the calculator keys, and the single-key column on the far left of the keyboard.

General function key: See "function key."

Hard disk: An optional part of the computer used to store programs and data. Hard disks come in two models: internal and external. An internal hard disk is contained in the processor unit. An external hard disk is attached to the processor unit with its own power cord.

Hardware: The physical parts of the computer.

Internal DOS command: A DOS command that is resident in memory at all times (e.g., DIR) as opposed to a command which must be loaded from disk, such as CHKDSK.

In-use light: A red light located by each floppy disk drive. The in-use light lights up each time the computer reads from or writes to the diskette in the drive. An in-use light also lights up when you turn on the power, indicating the drive into which you should insert the system diskette. (An in-use light is actually a light-emitting diode, or L.E.D.)

Keyboard: The computer's separate typewriter-style device on which you type commands and data.

Load: To transfer a program from a diskette or hard disk volume into the processor unit memory so that you can work with the program.

Logged drive: Default drive.

Memory: The computer's electronic storage area, containing programs and data ready for use by the computer.

Memory test message: The display shown while the computer checks the processor memory prior to loading the DOS. The message is a clock symbol (asking you to wait), the letter M, and a number. The number indicates memory size.

MS-DOS: One of the operating systems used with the computer.

Operating system: A program that manages the basic operations used by all programs. Examples of operating systems are CP/M-86 and MS-DOS.

Power-on message: The diskette-shaped symbol and flashing arrow shown when you turn on the power. The message tells you the computer is ready for you to insert a diskette.

Primary boot volume: The hard disk volume that you use to load the DOS.

Processor unit: One of the three main parts of the computer. The processor unit contains the floppy disk drives and the central processing unit (CPU), the "brains" of the computer.

Program: A set of instructions for a computer. Programs tell the computer how to do specific tasks.

Read: To retrieve data from a diskette. The computer can read programs and data files.

Reset: The act of pressing the reset button.

Reset button: The square button on the rear of the processor unit. When you press the reset button, the computer abandons the current file and/or program and tries to reload the DOS. If the DOS is not available to the computer, it displays the power on message.

Sign-on message: Identifying information the computer displays when a program is loaded.

Single-sided diskette: A diskette that can record information on one side only.

Single-sided drive: A floppy disk drive that can read or write only on a single side of a diskette.

Soft error: An error which may be caused by a worn or flawed section of a diskette or hard disk. You can check the number of soft errors on a diskette with FORMAT or DCOPY. Diskettes that contain over 2Ø soft errors are unreliable and should be discarded.

Software: Computer programs.

Specific function key: See "function key."

System diskette: A diskette containing an operating system.

Track: A circular section of a diskette. Your computer's diskettes contain 8Ø tracks on a side.

Typewriter keys: The main group of keys on the keyboard, arranged like a typewriter.

Utility: A program, provided with an operating system, used to help manage certain basic computer operations (such as FORMAT).

Volume: A section of the hard disk storage area. You use a hard disk volume much like you use a floppy disk drive that contains a diskette.

Write: To record or store data or programs on a diskette.

Write-protect: To protect a diskette, preventing the computer from writing over recorded data on it.

Write-protect notch: A notch on a diskette that you can cover with an adhesive tab to write-protect the diskette.

Zone: A group of diskette tracks. The drive motor runs at a different speed in each zone.