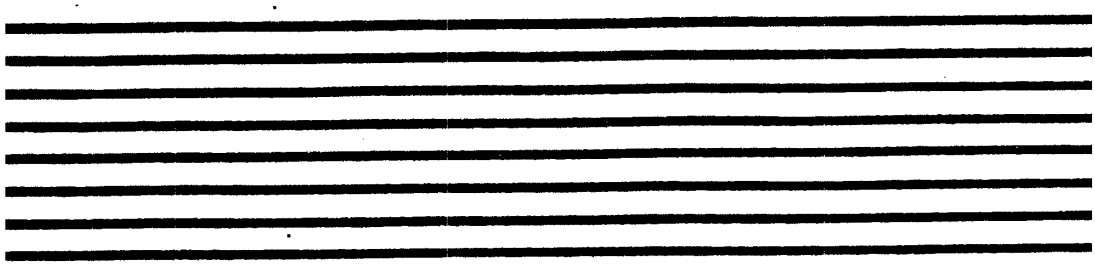
DIGITAL RESEARCH®

CP/M-8000™

Operating System

User's Guide



1038-2003-001

COPYRIGHT

Copyright © 1984 by Digital Research Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise without the prior written permission of Digital Research Inc., Post Office Box 579, Pacific Grove, California, 93950.

DISCLAIMER

Digital Research Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Digital Research Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation of Digital Research Inc. to notify any person of such revision or changes.

NOTICE TO USER

From time to time changes are made in the file names and in the files actually included with the distribution disk. This manual should not be construed as a representation or warranty that such files or facilities exist on the distribution disk or as part of the materials and programs distributed. Most distribution disks include a "READ.ME" file, which explains variations from the manual and which do constitute modification of the manual and the items included therewith. Be sure to read that file before using the software.

TRADEMARKS

CP/M and Digital Research and its logo are registered trademarks of Digital Research Inc. ASM, CP/M-8000, DDT, and TEX are trademarks of Digital Research Inc. Intel is a registered trademark of Intel Corporation. Olivetti is a registered trademark of Olivetti Corporation. Zilog is a registered trademark of Zilog, Inc. Z8000 is a trademark of Zilog, Inc.

The CP/M-8000 Operating System User's Guide was prepared using the Digital Research TEX™ Text Formatter and printed in the United States of America.

****
* First Edition: May 1984 *
****

Foreword

Welcome to the world of microcomputers opened to you by your Zilog® Z8000™ microprocessor. Welcome also to the world of application software accessible through your Digital Research CP/M-8000™ operating system. Digital Research designed CP/M-8000 especially for the Zilog Z8000 microprocessor which is the heart of your computer.

What CP/M-8000 Does For You

CP/M-8000 manages and supervises your computer's resources, including memory, console (screen and keyboard), printer, and communications devices. It also manages information stored magnetically on disks by grouping this information into programs or data files. CP/M-8000 can copy files from a disk to your computer's memory, or to a peripheral device such as a printer. To perform these functions, CP/M-8000 places various programs into your computer's memory and executes them according to commands you enter at your console.

Once in memory, a program runs through a set of steps that instruct your computer to perform a certain task. You can use CP/M-8000 to create your own programs, or you can choose from the wide variety of CP/M-8000 application programs that entertain, educate, and help solve your commercial and scientific problems.

What You Need to Run CP/M-8000 on Your Computer

The CP/M-8000 operating system runs on the Zilog Z8000 microprocessor. You need that microprocessor, a console device (generally a keyboard and display device such as a CRT screen), and at least one floppy disk drive. To use all the capabilities of CP/M-8000, you should have at least two disk drives. CP/M-8000 supports a hard disk subsystem, should a hard disk be part of your computer system.

CP/M-8000 and its utility programs are distributed on 5 1/4-inch floppy disks. The system disk contains the operating system and the most commonly used utility programs. Utility programs, or utilities, are application programs supplied with an operating system that perform housekeeping tasks, such as copying files or showing you the amount of free space on a disk.

The remaining disks contain additional utilities, plus the Unidot C compiler, the C language library, files that can be used to build versions of CP/M-8000 adapted to a variety of hardware environments, and sample Basic Input/Output Systems (BIOS).

CP/M-8000 Documentation Set

The CP/M-8000 documentation set includes three manuals:

- CP/M-8000 Operating System User's Guide (cited as CP/M-8000 User's Guide)
- CP/M-8000 Operating System Programmer's Guide (cited as CP/M-8000 Programmer's Guide)
- CP/M-8000 Operating System System Guide (cited as CP/M-8000 System Guide)

The CP/M-8000 User's Guide introduces you to the CP/M-8000 operating system and describes how to use it. The user's guide assumes that the version of CP/M-8000 on your distribution disk is ready to run on your computer. To use this manual, you must be familiar with the parts of your computer, know how to set it up and turn it on, and how to handle, insert, and store disks. This guide does not, however, assume you have great deal of experience with computers.

The CP/M-8000 Programmer's Guide presents information for application programmers who are creating or adapting programs to run under CP/M-8000. The programmer's guide includes information on the CP/M-8000 assembler and debugger that experienced programmers can use to create new CP/M-8000 programs.

The CP/M-8000 System Guide describes the procedures required to adapt CP/M-8000 for a variety of hardware environments.

How the User's Guide is Organized

This guide begins with simple examples, proceeds with basic concepts, and then presents a detailed reference section on CP/M-8000's commands. The first four sections describe CP/M-8000 operation for the first-time user.

Section 1 introduces CP/M-8000 and tells you how to start the operating system, enter commands, edit the command line, and create backup copies of your distribution disks. Section 2 discusses files, disks, and drives. Section 3 describes how to use CP/M-8000 to manage your printer and console.

Section 4 explains concepts that will help you to use CP/M-8000 commands to your best advantage. If you are new to CP/M®, read the first four sections carefully. They give you a general understanding of how to use the operating system before you proceed to the specific command descriptions.

Section 5 provides detailed information on CP/M-8000's utility programs. It is arranged alphabetically for easy reference. Section 6 tells you how to use ED, the CP/M-8000 file editor. With ED, you can create and edit program source, text, and data files.

Appendix A lists the messages CP/M-8000 displays when it encounters special conditions, and describes corrective actions where necessary. Appendix B lists the filetypes associated with CP/M-8000. Appendix C lists and defines the CP/M-8000 control characters. Appendix D explains the operation of a machine-specific FORMAT utility. The Glossary provides a list of commonly used computer terms.

Please note that boldface type following the computer prompt in command lines represents user input.

If you are new to computers, you might find some of the topics a bit difficult to understand at first. Learning to use your computer is a challenge, and we hope you'll find it fun. This book moves step-by-step so that you can quickly proceed from opening your new system disk package to using CP/M-8000's powerful facilities.

Contents

1 Introduction to CP/M-8000

How to Start CP/M-8000 1-1

Command Line 1-2

Why You Should Back Up Your Files 1-3

How to Make Copies of Your CP/M-8000 Disks 1-4

2 Files, Disks, and Drives

What is a File? 2-1

How to Create Files 2-1

Naming Files 2-2

Accessing Files Using the Correct Drive 2-3

Accessing Files Using the Correct User Numbers 2-4

Accessing More Than One File 2-5

File Attributes 2-6

How Files Are Stored on a Disk 2-6

Changing Floppy Disks 2-7

3 Console and Printer

Controlling Console Output 3-1

Controlling Printer Output 3-1

Console Line Editing 3-1

Assigning Logical Devices 3-3

4 CP/M-8000 Command Concepts

Two Kinds of Commands 4-1

Built-in Commands 4-1
Transient Utility Commands 4-2

How CP/M-8000 Searches for Program and Data Files . . . 4-4

Finding Files 4-4
Finding Program Files 4-5

Executing Multiple Commands 4-6

Terminating Programs 4-6

Parts of a File Specification 4-7

How Commands Are Described 4-10

5 Command Summary

COPY (Copy Disk) Command 5-2

DIR and DIRS Commands 5-6

ED (Character File Editor) Command 5-8

ERA Command 5-13

PIP Command 5-15

Single File Copy 5-15
Multiple File Copy 5-18
Combining Files 5-19
Copy Files to and from Auxiliary Devices 5-20
Multiple Command Mode 5-23
Using Options with PIP 5-23

REN Command 5-29

STAT Command 5-31

Free Space on Disk 5-32
Files - Display File Space and Access Mode 5-33
Set File Access Modes (Attributes) 5-36
Display Disk Status 5-37
Display User Numbers with Active Files 5-37
Display STAT Commands and Device Names 5-38
Display and Set
Physical-to-Logical Device Assignments 5-38

SUBMIT Command 5-40

Creating the SUB File 5-40
Executing the SUBMIT Command 5-42

TYPE Command 5-45

USER Command 5-47

6 CP/M-8000 Editor

Introduction to ED 6-1

Starting ED 6-1

ED Operation 6-2

Appending Text into the Buffer 6-4
ED Exit 6-5

Basic Editing Commands 6-6

Moving the Character Pointer 6-7
Displaying Memory Buffer Contents 6-9
Deleting Characters 6-10
Inserting Characters into the Memory Buffer 6-12
Replacing Characters 6-14

Combining ED Commands 6-14

Moving the Character Pointer 6-15
Displaying Text 6-15
Editing 6-16

Advanced ED Commands 6-17

Moving the CP and Displaying Text 6-17
Finding and Replacing Character Strings 6-18
Moving Text Blocks 6-22
Saving or Abandoning Changes: ED Exit 6-23

ED Error Messages 6-25

Appendixes

A Error Messages A-1

B Filetypes B-1

C CP/M-8000 Control Character Summary C-1

D FORMAT Command D-1

Glossary Glossary-1

Tables

3-1. CP/M-8000 Line-editing Control Characters 3-2
3-2. CP/M-8000 Logical Devices 3-3

4-1. Built-in Commands 4-2
4-2. Transient Utility Commands 4-3
4-3. Command Line Delimiters 4-9
4-4. CP/M-8000 Filetypes 4-9
4-5. Command Line Conventions 4-11

5-1. ED Command Summary 5-9
5-2. PIP Options 5-24

6-1. Text Transfer Commands 6-4
6-2. Basic Editing Commands 6-7
6-3. CP/M-8000 Line-editing Controls 6-12
6-4. ED Error Symbols 6-25
6-5. ED Disk File Error Messages 6-27

A-1. CP/M-8000 Error Messages A-1

B-1. Common Filetypes B-1

C-1. CP/M-8000 Control Characters C-1

Figures

6-1. Overall ED Operation 6-3

Section 1

Introduction to CP/M-8000

This section tells you how to start CP/M-8000, how to enter and edit a command line, and how to make backup copies of your CP/M-8000 distribution disks.

How to Start CP/M-8000

Starting or loading CP/M-8000 means causing your computer to read a copy of the operating system from your CP/M-8000 system disk into its memory. This is called a boot or cold start. Because the boot process varies from computer to computer, you must follow the manufacturer's instructions for the computer you use. The next two paragraphs describe a typical, but not universal, booting process.

First, turn your computer's power switch to its "on" position. Next, insert your CP/M-8000 system disk into the computer's initial drive and close the drive door. The initial drive is also the default drive, the drive CP/M-8000 goes to if no drive is specified. In this section, assume that the initial drive is A and the disk is removable. Now press the RESET or RESTART button. This automatically loads CP/M-8000 into your computer's memory.

If you want to restart CP/M-8000 at some point after initially loading the system, first make sure your CP/M-8000 system disk is in the initial drive and then press the RESET or RESTART button. This is called a system reset. It has the same effect as pressing RESET or RESTART when you first power up your computer. Performing a system reset erases any data previously loaded into your computer's memory. This means that any data not stored on a disk is lost.

After CP/M-8000 is loaded into memory, the following message is displayed on your screen:

```
CP/M-8000(tm)  Version n.n 03/17/84
Copyright(c) 1984 Digital Research Inc.
```

The version number, represented above by n.n, tells you the version of CP/M-8000 that you own. After this display, the following two characters appear on your screen:

```
A>
```

This is the CP/M-8000 system prompt. The system prompt tells you that CP/M-8000 is ready to read a command from your keyboard. In this example, the prompt also tells you that drive A is your default drive. This means that until you tell CP/M-8000 to do otherwise, it looks for program and data files on the disk in drive A.

Command Line

CP/M-8000 performs certain tasks according to specific commands that you type at your keyboard. A CP/M-8000 command line consists of a command keyword, an optional command tail, and a RETURN keystroke. The command keyword identifies a command (program) to be executed. The command tail can contain extra information for the command, such as a filename or parameters. To end the command line, you must press the RETURN key. Note that the RETURN key can be marked ENTER, RETURN, CR, or something similar on your keyboard. In this guide, RETURN refers to the RETURN key.

As you type characters at the keyboard, they appear on your screen. The single-character position indicator, called the cursor, moves to the right as you type characters. If you make a typing mistake, press either the BACKSPACE key (if your keyboard has one) or CTRL-H to move the cursor to the left and correct the error. CTRL is the abbreviation for the CONTROL key. To type a control character, hold down the CTRL key and press the required letter key. For example, to move the cursor to the left, hold down CTRL and press the H key.

You can type the keyword and command tail in any combination of uppercase and lowercase letters. CP/M-8000 treats all letters in the command line as uppercase.

Generally, you type a command line directly after the system prompt. However, CP/M-8000 does allow spaces between the prompt and the command keyword.

Let's use the DIR command to demonstrate how CP/M-8000 reads command lines. The DIR command tells CP/M-8000 to display a directory containing the names of the disk files on your screen. Type DIR after the system prompt and press RETURN.

```
A>DIR
```

CP/M-8000 responds to this command by writing the names of those files that are stored in the default storage area. For example, if you have your CP/M-8000 system disk in the default drive A, these filenames, among others, appear on your screen:

```
STAT      Z8K
PIP       Z8K
```

CP/M-8000 takes the command exactly as you type it and compares your input to executable program files in the CP/M-8000 directory. If the command matches a program file in the directory, CP/M-8000 executes that program. If the first word you type does not match any file in the directory, CP/M-8000 echoes this command and puts a question mark after it. For example, if you mistype the DIR command as DJR, you see the following:

```
A>DJR
DJR?
```

which tells you that CP/M-8000 cannot find that command. To correct simple typing errors, use the BACKSPACE key, or hold down the CTRL key and press H to move the cursor to the left. You cannot correct typing errors after you press the RETURN key. The command must then be typed correctly in its entirety. CP/M-8000 supports other control characters that help you edit command lines efficiently. Section 3 tells how to use control characters to edit command lines and other information you enter at your console.

The DIR command accepts a filename as a command tail. You can use DIR with a filename to see if a specific file is on the disk. For example, to check that the transient utility program COPY.Z8K is on your system disk, type

```
A>DIR COPY.Z8K
```

CP/M-Z8K performs this task by displaying either the name of the file you specified or the message "No File."

Be sure to type at least one space after DIR to separate the command keyword from the command tail. If you do not, CP/M-8000 responds as shown.

```
A>DIRCOPY.Z8K
DIRCOPY.Z8K?
```

Why You Should Back Up Your Files

Human or computer errors sometimes destroy valuable programs or data files. By mistyping a command, for example, you can accidentally erase a program that you just created or a data file that has been months in the making. A similar disaster can result from an electronic component failure.

Data processing professionals avoid losing programs and data by making copies of valuable files. Always make a working copy of any new program you purchase and save the original. If the program is accidentally erased from the working copy, you can easily restore it from the original.

It is also wise to make frequent copies of new programs or data files as you develop them. The frequency of making copies varies. As a general rule, make a copy at the point where it will take ten to twenty times longer to reenter the information than it will to make the copy.

Note: If you suspect a hardware problem has interfered with your copying, do not use your valuable backup disk to restart or continue the copy process. Hardware problems have even greater potential for causing harm to your disks than software problems do. If you need to locate a hardware problem, use noncritical disks.

So far, we have not discussed any commands that change information recorded on your CP/M-8000 system disk. Before we do, make a few working copies of your distribution disks.

How to Make Copies of Your CP/M-8000 Disks

To back up your CP/M-8000 disks, you need several floppy disks. The backup disks can be factory-fresh or used. You might want to format new disks or reformat used disks with the disk formatting program that accompanies your particular computer. If the disks are used, make sure that they do not contain any files that you want to save.

If your computer's manufacturer has provided a special program to copy disks, you can use that program to make backups. Otherwise, you can use the COPY utility program that comes with CP/M-8000. COPY is a track-by-track disk copying program that can copy the operating system loader and utility programs from your system disk. COPY creates an exact copy of your system disk. Before you can use COPY however, you probably have to format the disks that will become your backups. Formatting writes the proper density, sector size, and number of tracks onto disks so that they can be used with CP/M-8000. See Appendix D for a description of a machine-specific FORMAT utility.

To use the COPY utility, simply type COPY and press the RETURN key. You see the following message:

```
COPY VER 1.1
```

followed by a mode menu:

```
MODE            FUNCTION

ALL             Copy the whole disk
BOOT            Copy the boot tracks only
FILES           Copy the files only
END             End the program
```

Select the "ALL" mode for backing up your CP/M-8000 distribution disks by typing ALL and pressing RETURN. The following prompt appears on the screen:

Enter SOURCE drive: _

You must now enter a drive letter from A to P and press RETURN. Because your CP/M-8000 system disk is in drive A, press A, and then press RETURN. The program then prompts you to

Enter DESTINATION drive: _

Again, you must select a drive letter, and press RETURN. Then the program gives you a chance to abort the process by prompting:

(^C to ABORT)
RETURN to copy All from A to <destination>

The program gives you a chance to review what you selected before proceeding. If you still want to proceed, press the RETURN key. If not, hold down the control key while pressing C (^ means "CTRL" in the example above) and the process is terminated. For an expanded treatment of COPY see Section 5.

End of Section 1

Section 2

Files, Disks, and Drives

CP/M-8000's most important task is to access and maintain files on your disks. With CP/M-8000 you can create, read, write, copy, and erase disk files. This section tells you what a file is, how to create, name, and access a file, and how files are stored on your disks. It also tells how to change disks and change the default drive.

What is a File?

A CP/M-8000 file is a collection of related information stored on a disk. Every file on a given disk must have a unique name because CP/M-8000 uses that name to access that file. A directory is also stored on each disk. The directory contains a list of the files stored on that disk and the locations of each file on the disk.

In general, there are two kinds of files: program (command) files and data files. A program file is an executable file, a series of instructions the computer can follow step by step. A data file is usually a collection of information: a list of names and addresses, the inventory of a store, the accounting records of a business, the text of a document, or similar related information. For example, your computer cannot execute names and addresses, but it can execute a program that prints names and addresses on mailing labels.

A data file can also contain the source code for a program. Generally, a program source file must be processed by an assembler or compiler before it becomes a program file. In most cases, an executing program processes a data file. However, there are times when an executing program processes a program file. For example, the copy program, PIP, can copy one or more program files.

How to Create Files

There are many ways to create a file. You can create a file by copying an existing file to a new location, perhaps renaming it in the process. With CP/M-8000, you can use the PIP command to copy and rename files. A second way of creating a file is to use a text editor. The CP/M-8000 text editor ED, described in Section 6, can create a file and assign it the name you specify. Finally, some programs such as ASM™ create output files as they process input files.

Naming Files

CP/M-8000 identifies every file by its unique file specification. The simplest kind of file specification is a one- to eight-character filename, such as:

MYFILE

A file specification can have up to three parts: a drive specifier, a filename, and a filetype.

The drive specifier is a single letter (A through P) followed by a colon. Each drive in your system is assigned a letter. When you include a drive specifier as part of the file specification, you are telling CP/M-8000 that the file is stored on the disk currently in that drive. For example, if you enter

B:MYFILE

CP/M-8000 looks in drive B for the file MYFILE.

The filename can be from one to eight characters. When you create a filename, try to let the name tell you something about what the file contains. For example, if you have a list of customer names for your business, you could name the file

CUSTOMER

so that the name is eight or fewer characters, and gives you some idea of what is in the file.

As you begin to use your computer with CP/M-8000, you will find that files fall naturally into categories. To help you identify files belonging to the same category, CP/M-8000 allows you to add an optional one- to three-character extension, called a filetype, to the filename. When you add a filetype to the filename, you must separate the two with a delimiter, in this case a period. Try to use three letters that will tell you something about the file's category. For example, you could add the filetype NAM to the file that contains a list of customer names as follows:

CUSTOMER.NAM

When CP/M-8000 displays file specifications in response to a directory or DIR command, it adds blanks to short filenames so that you can compare filetypes quickly. Notice that the period delimiter does not appear on your screen.

The program files that CP/M-8000 loads into memory from a disk have different filetypes, but are in the category of Z8000 and equivalent programs that run with CP/M-8000. The filetypes Z8K and REL identify this category of executable programs.

We recommend that you create file specifications from letters and numbers. Because they have special meanings for CP/M-8000, you must not use the following characters in filenames or filetypes:

< > = , ! | * ? & / [ ] ( ) . : ; + - \

A complete file specification containing all possible elements consists of a drive specification, a primary filename, and a filetype, each separated from the others by its appropriate delimiters, as shown in the following example:

```
A:DOCUMENT.LAW
```

Accessing Files Using the Correct Drive

When you type a file specification in a command tail without a drive specifier, the built-in or transient utility looks for the file in the default drive, the drive named by the system prompt. For example, if you type the command

```
A>DIR COPY.Z8K
```

DIR looks in the directory of the disk in drive A for COPY.Z8K. If you have another drive, B for example, you need a way to tell CP/M-8000 to access the disk in drive B and not the disk in the default drive. For this reason, CP/M-8000 lets you precede a filename with a drive specifier. For example, in response to the command

```
A>DIR B:MYFILE.LIB
```

CP/M-8000 looks for the file MYFILE.LIB in the directory of the disk in drive B. When you give a command to CP/M-8000, you should note which disk is in the default drive. Many application programs require that the data files they access be stored in the default drive.

You can also precede a program filename with a drive specifier, even if you use the program filename as a command keyword. For example, if you type the following command:

```
A>B:PIP
```

CP/M-8000 looks in the directory of the disk in drive B for the file PIP.???. If CP/M-8000 finds PIP on drive B, it loads PIP into memory and executes it.

If you need to access many files on the same drive, you might find it convenient to change the default drive so that you do not need to enter a drive specifier repeatedly. To change the default drive, simply type the drive specifier (including the colon) next to the system prompt and press RETURN. In response, CP/M-8000 changes the system prompt to display the new default drive:

```
A>B:
B>
```

As you have seen, the drive specifier is an important part of the file specification. Moving a disk to a different drive changes the file specifications for the files on that disk. Please keep this in mind when moving your disks from one drive to another.

Section 4 presents more information on how CP/M-8000 locates program and data files.

Accessing Files Using the Correct User Number

CP/M-8000 further identifies all files by assigning each file a user number from 0 to 15 when the file is created. The user number for each file is recorded in the disk directory. User numbers allow you to separate your files into sixteen file groups. User numbers are particularly useful for organizing hard disk space.

When you use a CP/M-8000 utility to create a file, the file is assigned to the current user number, unless you use PIP to copy the file to another user number. You can determine the current user number by looking at the system prompt.

```
4A>             User number 4, drive A
A>              User number 0, drive A
2B>             User number 2, drive B
```

The user number always precedes the drive identifier in the system prompt. User 0, however, is the default user number and is not displayed in the prompt.

You can use the built-in command USER to change the current user number.

```
A>USER 3
3A>
```

Most commands can access only those files that have the current user number. For example, if the current user number is 7, a DIR command displays only the files in user 7. However, if an executable program file or a SUBMIT file resides in user 0, that file can be accessed from any user number. See Section 5 for more information on SUBMIT files.

Accessing More than One File

Certain CP/M-8000 built-in and transient utilities can select and process several files when special wildcard characters are included in the filename or filetype. A file specification containing wildcards can refer to more than one file because it gives CP/M-8000 a pattern to match. CP/M-8000 searches the disk directory and selects any file whose filename or filetype matches the pattern.

The two wildcard characters are ?, which matches any single letter in the same position and *, which matches any character at that position and any other characters remaining in the filename or filetype. The following list presents the rules for using wildcards.

- A ? matches any character in a name, including a space character.
- An * must be the last, or only, character in the filename or filetype. CP/M-8000 internally replaces an * with ? characters to the end of the filename or filetype.
- When the filename to match is shorter than eight characters, CP/M-8000 treats the name as if it ended with spaces.
- When the filetype to match is shorter than three characters, CP/M-8000 treats the filetype as if it ended with spaces.

Suppose, for example, you have a disk that contains the following six files:

A.Z8K, AA.Z8K, AAA.Z8K, B.Z8K, A.REL, and B.REL

The following wildcard specifications match all, or a portion of, the preceding files:

.	is treated as ????????.???
????????.???	matches all six names
*.Z8K	is treated as ????????.Z8K
????????.Z8K	matches the first four names
?.Z8K	matches A.Z8K and B.Z8K
?.*	is treated as ?.???
?.???	matches A.Z8K, B.Z8K, A.REL, and B.REL
A?.Z8K	matches A.Z8K and AA.Z8K

A*.Z8K	is treated as A???????.Z8K
A???????.Z8K	matches A.Z8K, AA.Z8K, and AAA.Z8K

Remember that CP/M-8000 uses wildcard patterns only while searching a disk directory. Therefore, wildcards are valid only in filenames and filetypes. You cannot use a wildcard character in a drive specifier.

File Attributes

When you create a file, CP/M-8000 gives it two attributes: DIR (for Directory) and R/W (for Read-Write). Through the STAT command, you can change these attributes from DIR to SYS (for System) and from R/W to R/O (for Read-Only).

The DIR and SYS attributes control whether CP/M-8000 displays the file's name in response to a DIR command or DIRS command. DIR displays only the file specifications having the DIR attribute; DIRS displays only the file specifications having the SYS attribute.

The second file attribute can be set to either R/W (Read-Write) or R/O (Read-Only). If a file is marked R/O, any attempt to write data to that file produces a Read-Only error message. Therefore, you can use the R/O attribute to protect important files. A file with the R/W attribute can be read or written to at any time. For more information on setting file attributes, see the STAT command in Section 5.

How Files Are Stored on a Disk

CP/M-8000 records the filename, filetype, user number, and attributes of each file in a special area of the disk called the directory. In the directory, CP/M-8000 also records which parts of the disk belong to which file. You can use the STAT command to determine the number of entries in your directory.

CP/M-8000 allocates directory and storage space for a file as records are added to the file. When you erase a file, CP/M-8000 reclaims storage in two ways: it makes the file's directory space available to catalog a different file, and it frees the file's storage space for later use. This handling of storage and directory space, called dynamic file allocation, is a powerful feature of CP/M-8000. You do not have to tell CP/M-8000 how big your file will become, because it automatically allocates more storage for a file as needed, and releases the storage for reallocation when the file is erased. Use the STAT command to determine how much space remains on the disk.

Changing Floppy Disks

CP/M-8000 cannot, of course, do anything to a file unless the disk that holds the file is inserted into a drive and the drive is ready. When a disk is in a drive, it is on-line and CP/M-8000 can access its directory and files.

At some time, you will need to take a disk out of a drive and insert another that contains different files. You can replace an on-line disk whenever you see the system prompt at your console. This is a clear indication that no program is reading from or writing to the disk in the drive.

You can also remove a disk and insert a new one when an application program prompts you to do so. This can occur, for example, when the data that the program uses does not fit on one floppy disk. Note that you must never remove a disk if a program is reading from or writing to it.

After you insert a floppy diskette into the drive, you must press CTRL-C. By pressing CTRL-C, you inform CP/M-8000 that all drives have been reset to the Read-Write state. This enables you to read and write to the new diskette. See "Terminating Programs" in Section 4 for more information on CTRL-C. Refer to the CP/M-8000 Programmer's Guide for more information on disk handling.

End of Section 2

Section 3

Console and Printer

This section describes how CP/M-8000 communicates with your console and printer. It shows you how to start and stop console and printer output, edit commands you enter at your console, and redirect console and printer input and output. It also explains the concept of logical devices and how they are used under CP/M-8000.

Controlling Console Output

Sometimes CP/M-8000 displays information on your screen too quickly for you to read it. An especially long display can scroll off the top of your screen before you have a chance to study it. You can tell the system to wait while you read the display by holding down the CTRL key and pressing S. A CTRL-S causes the display scrolling to stop. When you are ready for the display to continue scrolling, press CTRL-Q.

Controlling Printer Output

You can also use a control command to "echo" console output at the printer, if you have one. To start printer echo, enter a CTRL-P. To stop printer echo, enter another CTRL-P. While printer echo is in effect, any characters that appear on your screen are listed at your printer.

You can use printer echo with a DIR command to make a list of the files stored on a floppy disk. You can also use CTRL-P with CTRL-S and CTRL-Q to make a hard copy of part of a file. Use a TYPE command to start the display of the file at the console. See Section 5 for more detail of the STAT command. When the display reaches the part you need to print, press CTRL-S to stop the display, CTRL-P to enable printer echo, and then CTRL-Q to resume scrolling the display and start printing. You can use another CTRL-S, CTRL-P, CTRL-Q sequence to terminate printer echo.

Console Line Editing

You can correct simple typing mistakes in a command line with the BACKSPACE or CTRL-H key. CP/M-8000 also supports additional line-editing functions that you perform with control characters. You can use the control characters to edit command lines or input lines to most programs.

CP/M-8000 allows you to edit your command line using the set of characters listed in Table 3-1. To edit a command line in CP/M-8000, use control characters to delete characters to the left of the cursor, then replace them with new characters.

In the following example command line, the command keyword PIP is mistyped. The underbar (_) represents the cursor.

```
A>POP A:=B:*.*_
```

To move the cursor to the letter O, hold down the CTRL key and press letter H eleven times. CTRL-H deletes characters as it moves the cursor left, leaving the following command line:

```
A>P_
```

Now you can type the correct letters, press RETURN, and send the command to CP/M-8000.

```
A>PIP A:=B:*.*_
```

Table 3-1 lists all line-editing control characters for CP/M-8000.

Table 3-1. CP/M-8000 Line-editing Control Characters

Character	Meaning
CTRL-E	Forces a physical RETURN but does not send the command line to CP/M-8000. Moves the cursor to the beginning of the next line without erasing your previous input.
CTRL-H	Deletes a character and moves the cursor left one character position. Has the same function as backspace.
CTRL-I	Moves the cursor to the next tab stop. A tab stop is automatically set at each eighth column. Has the same effect as pressing the TAB key.
CTRL-J	Sends the command line to CP/M-8000 and returns the cursor to the left of the current line. Has the same effect as a RETURN or a CTRL-M.
CTRL-M	Sends the command line to CP/M-8000 and returns the cursor to the left of the current line. Has the same effect as a RETURN or a CTRL-J.
CTRL-R	Places a # at the current cursor location, moves the cursor to the next line, and displays any partial command you typed so far. Pressing RETURN sends the command to CP/M-8000.

Table 3-1. (continued)

Character	Meaning
CTRL-U	Discards all the characters in the command line, places a # at the current cursor position, and moves the cursor to the next command line.
CTRL-X	Discards all the characters in the command line and moves the cursor to the beginning of the current line.

You probably noticed that some control characters have the same meaning. For example, the CTRL-J and CTRL-M keystrokes have the same effect as pressing the RETURN key; all three send the command line to CP/M-8000 for processing. Also, CTRL-H has the same effect as pressing the BACKSPACE key.

Assigning Logical Devices

Most CP/M-8000 computer systems have a traditional console with a keyboard and screen display. Many also have letter-quality printers. If you use your computer for unusual tasks, you might want to add a different kind of character device to your system: a line printer, a teletype terminal, a modem, or even a joystick for playing games. To keep track of these physically different input and output devices, CP/M-8000 associates different physical devices with logical devices. When you receive your CP/M-8000 system, the logical devices are assigned to physical devices as shown in Table 3-2.

Table 3-2. CP/M-8000 Logical Devices

Logical Name	Device Type	Physical Assignment
CON:	Console input/ output	Keyboard/Screen
AXI:	Auxiliary input	Auxiliary input port (if one exists)
AXO:	Auxiliary output	Auxiliary output port (if one exists)
LST:	List output	Printer (if one exists)

Logical names are assigned to physical devices by the hardware manufacturer. However, in some implementations of CP/M-8000, you can change these assignments with the STAT command. For example, you can assign AXI and AXO to a modem so that your computer can communicate with others over the telephone. See STAT in Section 5, for a description of how to change logical name to physical device assignments.

End of Section 3

Section 4

CP/M-8000 Command Concepts

As discussed in Section 1, a CP/M-8000 command line consists of a command keyword, an optional command tail, and a RETURN keystroke. This section describes the two kinds of programs the command keyword can identify, and tells how CP/M-8000 searches for a program file on a disk. It also tells how to execute multiple CP/M-8000 commands, and how to reset the disk system.

Two Kinds of Commands

A command keyword identifies a program that resides either in memory as part of CP/M-8000, or on a disk as a program file. Commands that identify programs in memory are called built-in commands. Commands that identify program files on a disk are called transient utility commands.

Seven built-in commands and several transient utility commands are included with CP/M-8000. You can add utilities to your system by purchasing various CP/M-8000-compatible application programs. If you are an experienced programmer, you can also write your own utilities that operate with CP/M-8000.

Built-in Commands

Built-in commands are part of CP/M-8000 and come into memory when you boot the system. Because they reside in memory at all times, built-in commands have two advantages over transient commands:

- Built-in commands are always available regardless of the current user number or default drive displayed in the command prompt.
- Built-in commands start executing more quickly than the transient utilities because they do not have to be read from a disk.

Section 5 gives you the operating details of the built-in commands listed in Table 4-1.

Table 4-1. Built-in Commands

Command	Function
DIR	Displays filenames of files marked with the DIR (Directory) attribute in the directory.
DIRS	Displays filenames of files marked with the SYS (System) attribute in the directory.
ERA	Erases a file from the disk by removing the filename from the directory and by releasing the storage space occupied by the file.
REN	Renames a disk file.
SUBMIT	Executes a list of commands contained in a SUBMIT file that you create.
TYPE	Displays the contents of an ASCII (readable characters) file on your screen.
USER	Changes the current user number.

CP/M-8000 does not allow you to abbreviate the built-in commands.

Transient Utility Commands

A transient utility command executes a program that comes into memory only when you request it. When you enter a command that identifies a transient utility, CP/M-8000 loads the program file from the disk and passes it the filenames, data, or parameters you entered in the command tail. Because these programs are loaded from disk, the disk with the program file must be accessible to execute the command. How CP/M-8000 searches for command files is explained later in this section.

Section 5 provides the operating details for the most frequently used CP/M-8000 transient utilities listed in Table 4-2. Less frequently used utilities are described in the CP/M-8000 Programmer's Guide. Many of the utilities listed in Table 4-2 are programming tools that are of use only to systems and applications programmers.

Table 4-2. Transient Utility Commands

Name	Function	Where Explained
AR8K	Stores object files in the C run-time library: the Archive utility.	CP/M-8000 Programmer's Guide
ASZ8K	Invokes the assembler.	CP/M-8000 Programmer's Guide
COPY	Copies disks (including CP/M-8000 boot disks).	CP/M-8000 User's Guide, Section 5
DDT	Invokes Z8000 DDT, the CP/M-8000 debugger.	CP/M-8000 Programmer's Guide
DUMP	Displays a file in ASCII and hexadecimal formats.	CP/M-8000 Programmer's Guide
ED	Creates and alters character files.	CP/M-8000 User's Guide, Section 5, 6
FORMAT	Marks all disk sectors with the appropriate density and length.	CP/M-8000 User's Guide, Appendix D
LD8K	Invokes the Linker/Loader.	CP/M-8000 Programmer's Guide
NMZ8K	Invokes the NMZ8K utility that prints the symbol table.	CP/M-8000 Programmer's Guide
PIP	Copies, combines, or transfers specified files between peripheral devices.	CP/M-8000 User's Guide, Section 5
SIZEZ8K	Prints the size of a CP/M-8000 command file.	CP/M-8000 Programmer's Guide
STAT	Shows the access status for disks or files, the amount of free space on disks, the space occupied by files, or the logical-to-physical assignment of devices, according to options specified in the command line.	CP/M-8000 User's Guide, Section 5
XDUMP	Displays the header contents and symbol table of an object or command file.	CP/M-8000 Programmer's Guide

Table 4-2. (continued)

Name	Function	Where Explained
ZCC ZCC1 ZCC2 ZCC3	Compiler Command Interpreter Preprocessor Pass 1 Pass 2	The C Programming Language by B.W. Kernigham and D.M. Ritchie, Englewood Cliffs, NJ: Prentice Hall, 1978

How CP/M-8000 Searches for Program and Data Files

This section describes how CP/M-8000 searches for program and data files on disk. You need to understand the steps CP/M-8000 follows, especially if it appears that CP/M-8000 cannot find a program file you specified with a command keyword in a command line. If, for example, you have more than one disk drive on your system, the problem might be that CP/M-8000 is not looking on the drive where the file is stored.

Finding Files

As you recall, when you enter a command line, CP/M-8000 passes the command tail to the program identified by the command keyword. If the command tail contains a file specification without a drive specification, the program asks CP/M-8000 to search for the file on the default drive and in the current user number.

For example, if you enter the following command line:

```
3A>DIR MYFILE.TXT
```

CP/M-8000 limits its search to user 3, drive A. If MYFILE.TXT has the DIR attribute and exists in user 3 on drive A, CP/M-8000 confirms the file's existence by displaying the filename on your screen:

```
A:MYFILE.TXT
```

Note that the user number does not appear in the directory listing. DIR also reports "SYSTEM FILE(S) EXIST" if user 3 on drive A contains files marked with the SYS attribute.

If MYFILE.TXT has the SYS attribute and is stored in user 3 on drive A, DIR merely reports "SYSTEM FILE(S) EXIST" -- the filename, MYFILE.TXT, is not echoed to the screen.

Whether a file is marked SYS or DIR, the search for a file in a command tail does not extend beyond the current user number and current drive. If CP/M-8000 cannot find the file, the

program displays an error message at the console. Typically, this message is "File not found" or "No File". Which of these messages appears depends on the program identified by the command keyword.

Finding Program Files

When searching for a command file, CP/M-8000 uses an extensive search pattern. This search is based on your command keyword and the filetype, which you might or might not supply.

If you do add a filetype to your command keyword, CP/M-8000 first searches the current user number for the command file with that filetype. If it cannot find the file in the current user number, CP/M-8000 searches user 0 for the command file having the filetype you supplied.

If you omit a filetype from your command keyword, CP/M-8000 tries to supply one for you based on the following search pattern:

- CP/M-8000 first checks the current user number and drive to see if there is a program file with your command keyword as its name and Z8K as its filetype. If this search fails, CP/M-8000 then checks to see if it has a blank filetype. If the filetype is neither Z8K nor blank, CP/M-8000 then checks to see if the program file has a filetype of SUB.
- If the program file has none of these filetypes in the current user number, then CP/M-8000 checks user 0 for the same search pattern: first for the Z8K filetype, then for the blank filetype, and finally for the SUB filetype.
- At any point in the search process, CP/M-8000 stops the search if it finds the program file. CP/M-8000 then loads the program into memory and executes it. When the program terminates, CP/M-8000 displays the system prompt and waits for your next command.
- If CP/M-8000 does not find the command file, it repeats the command, follows it with a question mark, and waits for your next command.

If you include a drive specifier before the command keyword, CP/M-8000 first looks for the program file on the specified drive, in the current user number. If the program file is not found there, CP/M-8000 searches user 0 on the same drive. The search goes no further than this when a drive specifier is included in the command keyword.

Executing Multiple Commands

In the examples used so far, CP/M-8000 has executed only one command at a time. CP/M-8000 can also execute a list of commands. You can enter a list of commands at the system prompt, or you can put a frequently needed list of commands in a disk file. Once you have stored the list in a disk file, you can execute the list whenever you need to with a SUBMIT command.

To list multiple commands in a single command line, separate each command keyword and command tail from the next keyword with an exclamation point (!). When you complete the list, press RETURN. CP/M-8000 executes your commands in sequence, as shown below:

```
3A>DIRS!DIR EXAMP*.*!STAT A:
```

CP/M-8000 responds to the first of your three commands by reporting

```
NON-SYSTEM FILE(S) EXIST
```

Without waiting for any operator intervention, CP/M-8000 executes the second command:

```
3A>DIR EXAMP*.*
A: EXAMP7       : EXAMP1   TXT : EXAMP3       : EXAMP2   TXT
A: EXAMP4       : EXAMP5       : EXAMP6
```

Finally, CP/M-8000 executes the third command in the series:

```
3A>STAT A:

A: RW, Free Space:  98k
```

If you find yourself executing the same command list frequently, store the list in a disk file with a SUB filetype. See Section 5 for a description of using the SUBMIT command to repeatedly execute a list of commands.

Terminating Programs

There might be times when you want to terminate a running program before it terminates itself. You can terminate the DIR, DIRS, TYPE, and SUBMIT built-in commands and all transient utilities supplied with CP/M-8000 (whether invoked singly or in a multiple command line) by simply holding down the control key as you press the C key. In addition, a CTRL-C can terminate many of the application programs that you run with your CP/M-8000 operating system.

If you try to abort a program other than DIR, DIRS, TYPE, or SUBMIT, you might experience a delay between the time you press CTRL-C and the time the program terminates. This is because CP/M-8000 checks for CTRL-C only when reading from or writing to the console.

When you press CTRL-C during a program's execution, it causes a warm start, sometimes called a warm boot. Unlike other versions of CP/M, a warm start in CP/M-8000 does not reload the operating system from your disk back into your computer's memory; CP/M-8000 is always resident in memory. In CP/M-8000, a warm start does two things: it empties that portion of memory occupied by transient utilities and application programs, and it also resets all drives to a Read-Write state. At the end of a normal execution of a transient utility or a built-in command, CP/M-8000 performs a warm start.

Parts of a File Specification

This section describes the three parts of a file specification in a command line. To avoid confusion, each part is given a formal name. The three parts of a file specification are as follows:

- drive specifier is the optional disk drive A, B, C, ...P that contains the file or group of files to which you are referring. If a drive specifier is included in your command line, a colon must follow it.
- filename is the one- to eight-character first name of a file or group of files.
- filetype is the optional one- to three-character category name of a file or group of files. If the filetype is present, a period must separate it from the filename.

If you do not include a drive specifier, CP/M-8000 automatically uses the default drive. If you omit the period and the filetype, CP/M-8000 assumes that the filetype is either Z8K, three blanks, or SUB.

This general form is called a file specification. A file specification identifies a particular file or group of files in the directory of the on-line disk as indicated by the drive specifier. For example,

B:MYFILE.DAT

is a file specification that indicates drive B:, filename MYFILE, and filetype DAT. File specification is abbreviated to simply

filespec

in the command syntax statements in Section 5.

Some CP/M-8000 commands accept wildcards in the filename and filetype parts of the command tail. For example,

B:MY*.A??

is a file specification with drive specifier B:, filename MY*, and filetype A??. This file specification might match several files in the directory.

Put together, the parts of a file specification are represented like this:

d:filename.typ

where d: represents the optional drive specifier, filename represents the one- to eight-character filename, and typ represents the optional one- to three-character filetype. The syntax descriptions in this manual use the term filespec to indicate any valid combination of the elements included in the file specification. The following list shows valid combinations of the elements of a CP/M-8000 file specification.

- filename
- filename.typ
- d:filename
- d:filename.typ

The characters in Table 4-3 have special meaning in CP/M-8000, so do not use these characters in a file specification except as indicated.

Table 4-3. Command Line Delimiters

Character	Meaning
< > = , | ; & ( ) + - tab [ ] space RETURN keystroke	file specification delimiters
:	drive delimiter in file specification
.	filetype delimiter in file specification
< >	option list delimiters, reserved for future use
|	command separator
[ ]	option list delimiters for global and local options
;	comment delimiter at the beginning of a command line

CP/M-8000 has established several file groups. Table 4-4 lists some of their filetypes with a short description of each family. Appendix B provides the complete list.

Table 4-4. CP/M-8000 Filetypes

Filetype	Meaning
8KN	nonsegmented assembler source file
8KS	segmented assembler source file
Z8K	Z8000 (or equivalent machine language) executable program
C	C language source file
SUB	list of commands to be executed by SUBMIT
$$$	temporary file

In some of the command summaries, descriptive qualifiers are used in the syntax line with filespecs to further qualify the type of filespec accepted by the commands. For example, wildcard-filespec denotes wildcard specifications, dest-filespec denotes a destination filespec, and src-filespec denotes a source filespec.

How Commands Are Described

In Section 5, CP/M-8000 commands appear alphabetically. Section 5 also describes the notation that indicates the optional parts of a command line and other syntax notation. Each command is described according to the following format:

- The description begins with the command keyword in uppercase.
- The syntax section gives you one or more general forms to follow when you compose the command line.
- The explanation section defines the general use of the command keyword, and points out exceptions and special cases. The explanation sometimes includes tables or lists of options that you can use in the command line.
- The examples section lists a number of valid command lines that use the command keyword.

The notation in the syntax lines describes the general command form using these rules:

- Words in capital letters must be spelled as shown, but you can use any combination of upper- or lowercase letters.
- A lowercase word in the syntax line has a general meaning that is defined in the text.
- The symbolic notation d:, filename, typ, and filespec have the general meanings described in the preceding section.
- You must include one or more space characters where a space is shown, unless otherwise specified. For example, the PIP options do not need to be separated by spaces.

The following table defines the special syntactical symbols used in defining command line syntax. Unless otherwise noted, none of these symbols actually appear on your screen's command line; they exist solely on these pages so that you can see the general form of a given command line. Understanding this notation helps you form a valid command.

Table 4-5. Command Line Conventions

Symbol	Meaning
n	You can substitute a number for n.
o	You can substitute an option or a list of options.
s	You can substitute a string, which consists of a group of characters, for s.
{ }	Items within braces are optional. You can enter a command without the optional items. The optional items add effects to your command line.
[ ]	Items in square brackets are options or an option list. If you use an option specified within the brackets, the brackets must enclose the option. If the right bracket is the last character on the command line, it can be omitted. These brackets actually appear on your command line. They have the same meaning in a command line as they do in the command summary examples.
( )	Items in parentheses indicate a range of options. If you use a range from an option list, you must enclose the range within parentheses.
...	Ellipses tell you that the previous item can be repeated any number of times.
|	The "or" bar separates alternative items in a command line. You can select any or all of the alternatives specified. Mutually exclusive options are indicated in additional syntax lines or are specifically noted in the text.
↑ or CTRL	Represent the CTRL key on your keyboard. As a rule, control characters that do not have a command-line editing function appear on the screen with the circumflex or caret symbol (^) preceding the letter. Control characters that have command-line editing functions (see Table 3-1) do not appear on the screen when they are executed.
<cr>	Indicates a RETURN or carriage return. In most of the examples in this book, the carriage return is implied and therefore does not appear in the examples.

Table 4-5. (continued)

Symbol	Meaning
*	Wildcard character that replaces all or part of a filename or filetype. Appears on your screen and has the same meaning as it does in the command summary examples.
?	Wildcard character that replaces any single character in the same position of a filename or filetype. Appears on your screen and has the same meaning as it does in the command summary examples.

Let's look at some examples of syntax notation. The CP/M-8000 DIR (Directory) command displays the names of files cataloged in the disk directory.

The syntax of the DIR command shows how to use the command line syntax notation:

```
Syntax:        DIR [d:] | [filespec]
                    |         |
                 optional  optional
```

This tells you that the command tail following the command keyword DIR is optional. In other words, DIR alone, DIR with a filespec, DIR with just a drive specifier, or DIR with a drive specifier and a filespec are all valid versions of the DIR command. For example,

```
DIR
DIR JOHN.LTR
DIR B:
DIR B:JOHN.LTR
DIR B:JOHN.*
DIR B:*.LTR
DIR B:*.*
```

are valid commands. The last example, incidentally, is equivalent to DIR b:, as shown by the general syntax:

```
DIR [d:] | [filespec]
```

These are the only forms DIR can take with respect to JOHN.LTR, any file, or no file. Understanding the syntax lets you see all the possibilities.

The CP/M-8000 command PIP (Peripheral Interchange Program) provides an example of a different kind of syntax. PIP can copy files from the disk to the screen or printer. PIP can combine two or more files into one longer file. PIP can also rename files after copying them. All of this functionality can be represented in the syntax:

PIP dest-filespec=src-filespec{,filespec...}

In this example, dest-filespec is further defined as a destination file specification or peripheral device (printer, for example) that receives data. Similarly, src-filespec is a source file specification or peripheral device (keyboard, for example) that transmits data. PIP accepts wildcards in the filename and filetype. See the PIP command in Section 5 for details regarding other capabilities of PIP. There are, of course, many valid command lines that come from this syntax. Some of them are shown below.

```
A>PIP NEWFILE.DAT=OLDFILE.DAT
A>PIP A:THISFILE.DAT=B:THATFILE.DAT
A>PIP B:=A:THISFILE.DAT
A>PIP B:X.BAS=Y.BAS, Z.BAS
A>PIP X.BAS=A.BAS, B.BAS, C.BAS
A>PIP B:=A:*.BAK
A>PIP B:=A:*.*
```

End of Section 4

Section 5

Command Summary

This section alphabetically presents the commands and programs supplied with your CP/M-8000 operating system. As described in Section 4, each command summary contains a short explanation of the command's operation and example command lines. ED, the CP/M-8000 editor, is described in detail in Section 6. Other commands, such as ASZ8K and Z8000 DDT™, are described fully in the CP/M-8000 Programmer's Guide.

COPY (Copy Disk) Command

Syntax: COPY [option list]

Explanation: COPY copies the contents of one disk onto another disk. It does so on a track-by-track basis, giving you a literal image of the disk or selective part of the disk you want duplicated.

Unlike PIP, which needs to know both the type and length of file to be copied, COPY is not sensitive to the types of files or their lengths. Even so, COPY requires that your destination disk have the same format as your source disk. You might have to format a destination disk before using COPY. Consult the documentation for your computer's FORMAT utility for details.

As shown in the above syntax, you can type copy and then press the RETURN key. This is the simplest form of executing COPY. It is also the most interactive. When you load COPY this way, the following prompt comes to your screen:

Copy Ver 1.1

A mode menu then appears:

MODE	FUNCTION
ALL	Copy the whole disk
BOOT	Copy the boot tracks
FILES	Copy the non-boot tracks
END	End this program

Choose one mode from the mode menu by entering your choice. For example, if you want to copy everything on one of your CP/M-8000 disks, enter ALL to copy CP/M-8000 and all files beyond the boot tracks. Enter BOOT if you want to copy only the boot loader for CP/M-8000. Enter FILES if you want to copy only the non-boot tracks. Enter END if you want to leave the COPY program and return to the CP/M-8000 system prompt.

Once you have selected your mode, COPY prompts you with

Enter SOURCE drive: _

At the cursor position, type a drive letter in the range a through p and press the RETURN key.

Similarly, you see the prompt:

```
Enter DESTINATION drive: _
```

Reply with a drive letter in the range A through P and press the RETURN key. COPY prompts:

```
(^C to ABORT)
RETURN to copy <mode> from <source> to <destination>.
```

Mode, source, and destination have actual values on your screen in place of the words shown here. If these are the values you truly want COPY to use, press the RETURN key. COPY begins copying at this point, as indicated by the message,

```
***Copying nn Tracks***
```

that appears at the bottom of your screen. The nn in this prompt stands for the actual track numbers currently being copied.

If you decide to abort the copy process, type CTRL-C (that is, hold down the Control key and press the letter C). If you have not used the [A] (for automatic) option when you were at the CP/M-8000 command line, then you are offered a chance to do some more copying when COPY finishes your first copy. COPY saves the parameters you gave it for the last copy just made; then COPY asks you:

```
Do you wish to repeat the copy? _
```

Type Y if you do, N if you do not. If you type N, the COPY program ends without any further processing and returns you to the CP/M-8000 system prompt. If you type Y, COPY shows you the values used in the last copy session and invites you to start another copy session by entering a carriage return. For example, if you chose mode ALL, source drive A, and destination drive B in your initial copy session, COPY returns the following prompt for all subsequent copy sessions:

```
(^C to ABORT)
RETURN to copy ALL from A to B
```

If you decide at this point not to begin a new session, type CTRL-C. CTRL-C returns you to the CP/M-8000 system prompt.

Instead of typing COPY and pressing the RETURN key at the prompt, you can supply your copy options before COPY is loaded and executed. However, if you take advantage of this shortcut, you must present your options in a certain order. The order for COPY options is:

```
[mode option] before
[source drive option] before
[destination drive option] before
[automatic carriage return option]
      and/or [verify option] before
[manual carriage return]
```

Examples: Some representative combinations of the COPY command line appear below. With the exception of the A and V options, which are discussed below, each of the options in the above examples is explained in the preceding basic, interactive COPY session.

```
COPY ALL
COPY ALL A
COPY ALL A B
COPY ALL A B [A]
COPY ALL A B [V]
COPY ALL A B [AV]

COPY A
COPY B
COPY [A]
COPY [V]
COPY [AV]

COPY A B
COPY A B [A]
COPY A B [V]
COPY A B [AV]
```

Note: Be sure the source drive appears to the left of the destination drive in your option list. If you inadvertently get the drive letters of your destination drive and source drive turned around in this option list, you will copy your destination disk (probably blank) on to your source disk, thereby wiping it out. To avoid this, always enter the source drive letter before you specify the destination drive letter, then recheck what you've typed before pressing the RETURN key.

The [A] option acts as a virtual RETURN key at run-time. You usually would be required to set the copy process in motion once the parameters have been supplied to COPY. Specifying [A] allows the copy process to execute without interruption and without further operator interaction.

The [V] option invokes COPY's verification option. When you use the [V], COPY compares the data on source and destination disks following a copying operation to verify that the data on the two disks are the same. COPY with [V] runs more slowly than COPY without [V]. This is because [V] demands that COPY go through extra steps in reading the destination disk and comparing its data to the data on the source disk.

If you enter more options in the option list than COPY uses, those extra options are thrown away. For example, if you type

```
A>COPY FILES A B HELLO THERE
```

the COPY utility loads and then displays the message:

```
Extraneous argument ignored:  hello
Extraneous argument ignored:  there
(^C to ABORT)
RETURN to copy FILES from A to B _
```

Thus, COPY helps salvage the copy session for you by allowing you to either abort COPY or else continue with the copy process, using the only three valid options it found from the command line in this example.

DIR and DIRS Commands

Syntax:

```
DIR  [d:]
DIR  [filespec]

DIRS [d:]
DIRS [filespec]
```

Explanation: The DIR and DIRS built-in commands display the names of files contained in the directory of an on-line disk.

The DIR command lists the names of files in the current user number that have the Directory (DIR) attribute. The DIRS command displays the names of files in the current user number that have the System (SYS) attribute.

All executable files in user 0, regardless of their SYS or DIR attribute, can be executed in any user number. However, you must be in user 0 to display those files with DIR or DIRS.

If the drive and file specifications are omitted, the DIR command displays the names of all files with the DIR attribute on the disk in the default drive and current user number. In the same way, DIRS displays the SYS files.

If the drive specification is included, but the filename and filetype are omitted, the DIR command displays the names of all DIR files in the current user number on the disk in the specified drive. DIRS displays the SYS files.

Both DIR and DIRS accept wildcard filenames and filetypes. When you use wildcard characters, all filenames that satisfy the match are displayed on the screen.

If no filenames match the file specification, or if no files are cataloged in the directory of the disk in the named drive and user number, the DIR and DIRS commands each display the message:

```
No File
```

If SYS files that reside on the specified drive and user number match the file specification, DIR displays the message:

SYSTEM FILE(S) EXIST

If DIR files match the file specification, DIRS displays the message:

NON-SYSTEM FILES(S) EXIST

The SYS and DIR attributes are described under STAT later in this section.

Examples: The following example displays all DIR files cataloged in user 0 on the default drive A.

```
A>DIR
```

The following command displays all DIR files for user 0 on drive B.

```
A>DIR B:
```

The next command displays the name X.BAS if the file X.BAS is present for user 0 on drive B.

```
A>DIR B:X.BAS
```

The next command displays all DIR files with filetype BAS for user 4 on drive A.

```
4A>DIR *.BAS
```

The following command displays all DIR files for user 0 on drive A whose filename begins with the letter X, and whose three character filetype contains the first character C and last character D.

```
B>DIR A:X*.C?D
```

The next command displays all files for user 0 on drive A that have the SYS attribute.

```
A>DIRS
```

This wildcard form of the DIRS command displays all SYS files with the Z8K filetype on the default drive A for user 3.

```
3A>DIRS *.Z8K
```

ED (Character File Editor) Command

Syntax: ED input-filespec [d: | output-filespec]

Explanation: The ED transient utility lets you create and edit a disk file.

The ED utility is a line-oriented context editor. This means that you create and change character files line-by-line, or by referencing individual characters within a line.

The ED utility lets you create or alter the file named in the file specification. Please refer to Section 6 for a detailed description of the ED utility.

The ED utility uses a portion of your computer's memory as the active text buffer, a kind of temporary storage area where you can add, delete, or alter the characters in the file. Use the A command to read (Append) all or a portion of the file into the buffer. Use the W command when you want to write a portion of the file's contents from the buffer back into the file. The E command is used to write the entire contents the buffer back into the file and end the ED session.

An imaginary character pointer, called CP, is at the beginning of the buffer, between two characters in the buffer, or at the end of the buffer.

You interact with the ED utility in either command or insert mode. ED displays the * prompt on the screen when ED is in command mode. When the * appears, you can enter the single letter command that reads text from the buffer, moves the CP, or changes the ED mode of operation. When in command mode, you can use the line-editing characters CTRL-E, -H, -U, -X, and RUBOUT to edit your input. In insert mode, however, you can use only CTRL-H, -U, -X, and RUBOUT.

Table 5-1. ED Command Summary

Command	Action
nA	Append n lines from original file to memory buffer.
0A	Append file until buffer is one half full or until end of input is reached.
#A	Append file until buffer is full (or end of file).
B, -B	Move CP to beginning (B) or bottom (-B) of buffer.
nC, -nC	Move CP n characters forward (C) or back (-C) through buffer.
nD, -nD	Delete n characters before (-D) or after (D) the CP.
E	Save new file and return to CP/M-8000.
Fstring{↑Z}	Find character string.

Table 5-1. (continued)

Command	Action
E	Save the new file, then reedit, using the new file as the original file.
I	Enter insert mode; use ↑Z to exit insert mode.
Istring[↑Z]	Insert string at CP.
Jsearch_str^Zins_str^Zdel_to_str[↑Z]	Juxtapose strings. Juxtaposition does not support a repetitive execution of juxtapose on a single command line.
nK, -nK	Delete (kill) n lines from the CP.
nL, -nL, 0L	Move CP n lines.
nMcommands	Execute commands n times.
n, -n	Move CP n lines and display that line.
n:	Move to line n.

Table 5-1. (continued)

Command	Action
:ncommand	Execute command through line n.
Nstring{↑Z}	Extended find string.
O	Return to original file.
nP, -nP	Move CP n "pages" forward or backward (-P) in the buffer and display n pages on the console. A page equals 23 lines.
Q	Abandon new file, return to CP/M-8000.
R	Read X$$$$$$$.LIB file into buffer.
Rfilespec{↑Z}	Read filespec into buffer.
Sdelete string^Zinsert string{↑Z}	Substitute string.
nT, -nT, 0T	Type n lines.

Table 5-1. (continued)

Command	Action
U, -U	Uppercase translation.
V, -V, OV	Line numbering on/off, display free buffer space.
nW	Write n lines to new file.
nX	Write or append n lines to X$$$$$$$.LIB.
nXfilespec{↑Z}	Write n lines to filespec or append if previous X command applied to the same file.
0X	Delete file X$$$$$$$.LIB.
0Xfilespec{↑Z}	Delete filespec.
nZ	Wait n seconds.

ERA Command

Syntax: ERA [filespec]

Explanation: The ERA command removes one or more files from the directory of a disk. Wildcard characters are accepted in the filespec. Directory and data space are automatically reclaimed for later use by another file.

Use the ERA command with care because all files that satisfy the file specification are removed from the disk directory.

Command lines with the form:

ERA [d:]wildcard-filespec

require your confirmation because they erase an entire group of files, not just one file. The system will prompt you for confirmation with the following message:

Confirm (Y/N)?

Respond with Y if you want to remove all matching files, and N if you want to avoid erasing any files.

There is one instance in which the ERA command with a wildcard does not cause the system to prompt you. In a SUBMIT file, when you use a wildcard with an ERA command, CP/M-8000 does not ask for confirmation. Please see SUBMIT later in this section.

If no files match the file specification, you see the following message:

No File

Examples: This command remove the file X.PAS from the disk in user 0 on drive A.

```
A>ERA X.PAS
```

To delete all file with the filetype PRN from the disk in drive A, use 0, enter the following command. You see the system equest confirmation:

```
A>ERA *.PRN

Confirm (Y/N)?Y
```

The next example re ests that all files on drive B, user 5, be removed rom the disk.

```
5A>ERA B:*.*

Confirm (Y/N)?Y
```

PIP Command

Syntax: PIP dest-file{[Gn]}|dev=src-file{[o]}|dev{[o]}

Explanation: The PIP utility copies one or more files from one disk and/or user number to another. PIP can rename a file after copying it. PIP can combine two or more files into one file. PIP can also copy a character file from disk to the printer or other auxiliary logical output device. PIP can create a file on disk from the console or other logical device input. PIP can transfer data from a logical input device to a logical output device. Hence the name Peripheral Interchange Program.

Single File Copy

Syntax: PIP d:{[Gn]} = source-filespec{[options]}

PIP dest-filespec{[Gn]} = d:{[options]}

PIP dest-filespec{[Gn]} = source-filespec{[o]}

Explanation: The first form shows the simplest way to copy a file. PIP looks for the file named by source-filespec on the default or optionally specified drive. PIP copies the file to the drive specified by d: and gives it the same name as source-filespec. You can use the [Gn] option to place your destination file (dest-filespec) in the user number specified by n. The only option recognized for the destination file is [Gn]. Several options can be combined together for the source file specification (source-filespec). PIP's options are fully described below.

The second form is a variation of the first. PIP looks for the file named by dest-filespec on the drive specified by d:, copies it to the default or optionally specified drive, and gives it the same name as dest-filespec.

The third form shows how to rename the file after you copy it. You can copy it to the same drive and user number, or to a different drive and/or user number. Rules for options are the same. PIP looks for the file specified by source-filespec, copies it to the location specified in dest-filespec, and gives it the name indicated by dest-filespec.

Remember that PIP always goes to and gets from the current user number unless you specify otherwise with the [Gn] option.

Before you start PIP, be sure that you have enough free space in kilobytes on your destination disk to hold the entire file or files that you are copying. Even if you are replacing an old copy on the destination disk with a new copy, PIP still needs enough room for the new copy before it deletes the old copy. See the STAT command in this section.

Data is first copied to a temporary file to ensure that the entire data file can be constructed within the space available on the disk. PIP gives the temporary file the filename specified for the destination, with the filetype $$$. If the copy operation is successful, PIP changes the temporary filetype $$$ to the filetype specified in the destination.

If the copy operation succeeds and a file with the same name as the destination file already exists, the old file with the same name is erased before renaming the temporary file.

File attributes (SYS, DIR, RW, RO) are transferred with the files.

If the existing destination file is set to Read-Only (RO), PIP asks you if you want to delete it. Answer Y or N. Use the W option to write over Read-Only files.

You can include PIP options following each source name (see "PIP Options" below). There is one valid option, [Gn] (go to user number n), for the destination file specification. Options are enclosed in square brackets. Several options can be included for the source files. They can be packed together or separated by spaces. Options can verify that a file was copied correctly, allow PIP to read a file with the SYS attribute, cause PIP to write over Read-Only files, cause PIP to put a file into or copy it from a specified user number, transfer from lower- to uppercase, and much more.

Examples: Both of the following forms of this command cause PIP to read the file OLDFILE.DAT from drive A and put an exact copy of it onto drive B. This is called the short form of PIP, because the source or destination names only a drive and does not include a filename. When using this form, you cannot copy a file from one drive and user number to the same drive and user number. You must put the destination file on a different drive or in a different user number. See the section on PIP Options, and the section on the USER command. The second short form produces exactly the same result as the first one. PIP simply looks for the file OLDFILE.DAT on drive A, the drive specified as the source.

```
A>PIP B:=A:OLDFILE.DAT

A>PIP B:OLDFILE.DAT=A:
```

The following command copies the file OLDFILE.DAT from drive A to drive B and renames it to newfile.dat. The file remains as OLDFILE.DAT on drive A. This is the long form of the PIP command, because it names a file on both sides of the command line.

```
A>PIP B:NEWFILE.DAT=A:OLDFILE.DAT
```

Using this long form of PIP, you can copy a file from one drive and user number (usually user 0 because CP/M-8000 automatically starts out in user 0, the default user number) to the same drive and user number. This effectively gives you two copies of the same file on one drive and user number, each with a different name.

```
A>PIP NEWFILE.DAT=OLDFILE.DAT
```

The following command copies the file PROGRAM.DAT from user 1 on drive A to user 0 on drive B and renames the filetype on drive B, user 0, to BAK.

```
A>PIP B:PROGRAM.BAK = A:PROGRAM.DAT[G1]
```

In the next command, PIP copies the file named PROGRAM1.DAT on drive A and echoes [E] the transfer to the console, verifies [V] that the two copies are exactly the same, and gets [G3] the file program1.dat from user 3 on drive A. Because there is no drive specified for the destination, PIP automatically copies the file to the default user number and drive, in this case drive B, user 0.

```
B>PIP PROGRAM2.DAT=A:PROGRAM1.DAT[E V G3]
```

Multiple File Copy

Syntax: PIP d:[[Gn]] = [d:]wildcard-filespec[[o]]

Explanation: When you use a wildcard in the source specification, PIP copies qualifying files one-by-one to the destination drive, retaining the original name of each file. PIP displays the message, COPYING, listing each filename as the copy operation proceeds. PIP issues an error message and aborts the copy operation if the destination drive and user number are the same as those specified in the source.

Examples: The following command causes PIP to copy all the files on drive A with the filetype CMD to drive B.

```
A>PIP B:=A:*.CMD
```

The next command causes PIP to copy all the files on drive A to drive B. You can use this command to make a backup copy of your distribution disk. Note, however, that this command does not copy the CP/M-8000 bootstrap loader from the system tracks. COPY, or any other track-to-track copy program, copies the system tracks for you.

```
A>PIP B:=A:*.*
```

The command below causes PIP to copy all files beginning with PROG and having any filetype from drive A to drive B.

```
A>PIP B:=A:PROG????.*
```

The following command causes PIP to copy all the files with a filetype of BAS on drive A in the default user number, user 0 in this case, to drive B in user number 1. Remember that the DIR, TYPE, ERA and other commands only access files in the same user number from which they were invoked. See the USER command.

```
A>PIP B:[G1]=A:*.BAS
```

Combining Files

Syntax: PIP dest-file{[Gn]}=src-file{[opt]},file {[opt]}{,file{[opt]}...}

Explanation: This form of the PIP command lets you specify two or more files in the source. PIP copies the files specified in the source from left to right and combines them into one file with the name indicated by the destination file specification. This procedure is called file concatenation. You can use the [Gn] option after the destination file to place it in the user number specified by n. You can specify one or more options for each source file.

Most of the options force PIP to copy files character by character. In these cases PIP looks for a CTRL-Z character to determine the end of the file. All of the PIP options force a character transfer except the following:

Gn, K, O, R, V, and W.

Copying data to or from logical devices also forces a character transfer.

During character transfers, you can terminate a file concatenation operation by pressing CTRL-C.

When concatenating files, PIP searches only the last record of a file for the CTRL-Z end-of-file character. However, if PIP is doing a character transfer, it stops when it encounters a CTRL-Z character.

Use [O] if you are concatenating machine code files. [O] causes PIP to ignore an embedded CTRL-Z (end-of-file) character, which usually indicates the end-of-file character in a file.

Examples: The following three files named FILE1, FILE2, and FILE3 are joined from left to right and copied to NEWFILE.$$$. NEWFILE.$$$ is renamed to NEWFILE upon successful completion of the copy operation. All source and destination files are on the disk in the default drive A.

```
A>PIP NEWFILE=FILE1,FILE2,FILE3
```

In the next example, the file Y.ASM on drive A is joined with Z.ASM from drive B and placed in the temporary file X.$$$ on drive B. The file X.$$$ is renamed to X.ASM on drive B when PIP runs to successful completion.

```
A>PIP B:X.ASM = Y.ASM, B:Z.ASM
```

Copy Files to and from Auxiliary Devices

Syntax:

```
PIP dest-filespec {[Gn]} = source-filespec {[o]}
    AXO:                   AXI: {[options]}
    CON:                   CON: {[options]}
    PRN:                   NUL:
    LST:                   EOF:
```

Explanation: This form is a special case of the PIP command line that lets you copy a file from a disk to a device, from a device to a disk, or from one device to another. The files must contain printable characters. Each peripheral device is assigned to a logical name that identifies a source device that can transmit data, or a destination device that can receive data. A colon follows each logical device name so it cannot be confused with a filename. Press CTRL-C to abort a copy operation that uses a logical device in the source or destination.

The logical device names are listed as follows:

CON: Console is the physical device assigned to CON. When used as a source, usually the keyboard; when used as a destination, usually the screen.

AXI: Auxiliary Input or Output Device.

AXO: Auxiliary Output Device.

LST: The destination device assigned to LST, usually the printer.

There are three device names that have special meaning:

NUL: A virtual source device that produces 40 hexadecimal zeros.

EOF: A virtual source device that produces a single CTRL-Z, the CP/M-8000 end-of-file mark.

PRN: The printer device with tab expansion to every eighth column, line numbers, and page ejects every 60th line.

Examples:

In the following example, characters are first read from the console input device, generally the keyboard, and sent directly to your printing device. You type a CTRL-Z character to tell PIP that keyboard input is complete. At that time, PIP continues by reading character data from the file MYDATA.DAT on drive B. Because PRN: is the destination device, tabs are expanded, line numbers are added, and page ejects occur every 60 lines.

```
B>PIP PRN:=CON:,MYDATA.DAT
```

If CRT: is assigned to CON:, whatever you type at the console is written to the file NOTES.GEE on drive B. You must press the RETURN key and the line-feed key (or CTRL-J) to commit your input to the file. End the keyboard input by typing a CTRL-Z. Corrections of previously typed material are not permitted in this option.

```
A>PIP B:NOTES.GEE=CON:
```

If CRT: is assigned to CON:, as in the following example, whatever you type at the keyboard is written to the list device, generally the printer. Terminate input with a CTRL-Z.

```
A>PIP LST:=CON:
```

In the following example, the file DRAFT.TXT on drive B is written to the printing device. Any tab characters are expanded to the nearest column that is a multiple of 8.

```
A>PIP LST:=B:DRAFT.TXT[T8]
```

The command below causes PIP to write the file DRAFT.TXT to the list device. It automatically expands the tabs, adds line numbers, and ejects pages after sixty lines.

```
A>PIP PRN:=B:DRAFT.TXT
```

The PIP options S (Start) and Q (Quit) search only for uppercase character strings when you enter PIP from the CP/M-8000 command line. This is because the CCP automatically converts the command line to uppercase. If you want to search for a lowercase string, you must use PIP's interactive mode.

The following command causes NOTE.TXT to list at the console, CON: starting at the first occurrence of BEGINNING.

```
A>PIP CON:=NOTE.TXT[S BEGINNING^Z]
```

The CCP converts beginning to BEGINNING, if you enter the following command line.

```
A>PIP CON:=NOTE.TXT[S beginning^Z]
```

Because PIP cannot find BEGINNING in the file NOTE.TXT, you receive the following message:

```
ERROR:  START NOT FOUND
```

To avoid seeing this message, enter the Search option at the PIP prompt.

```
A>PIP
*CON:=NOTE.TXT[S beginning^Z]
```

Also use PIP's interactive mode when specifying Quit strings with the Q option.

Multiple Command Mode

Syntax: PIP

Explanation: This form of the PIP command starts the PIP utility and lets you type multiple command lines while PIP remains in user memory.

PIP writes an asterisk (*) on your screen when ready to accept input command lines.

You can type any valid command line described under previous PIP formats following the asterisk prompt.

Terminate PIP by pressing the RETURN key following the asterisk prompt. The empty command line tells PIP to discontinue operation and return to the CP/M-8000 system prompt.

Examples: The following command loads the PIP program. The PIP command input prompt (*) tells you that PIP is ready to accept commands. The effects of this sequence of commands are the same as shown in the previous examples, where the command tail is included in the command line. PIP is not loaded into memory for each command.

```
A>PIP
*MENFILE=FILE1,FILE2,FILE3
*APROG.CMD=BPROG.CMD
*A:=B:X.ASM
*B:=*.*
*
```

Using Options with PIP

Explanation: Options enable you to process your source file in special ways. You can expand tab characters, translate from upper to lowercase, extract portions of your text, verify that the copy is correct, and much more.

PIP options are listed below, using n to represent a number and s to represent a sequence of characters terminated by a CTRL-Z. An option must immediately follow the file or device it affects. The option must be enclosed in square brackets, []. For those options that require a numeric value, no blanks can occur between the letter and the value.

You can include the [Gn] option after a destination file specification. You can include a list of options after a source file or source device. An option list is a sequence of single letters and numeric values that are optionally separated by blanks and enclosed in square brackets, [].

Table 5-2. PIP Options

Option	Function
Dn	Delete any characters past column n. This parameter follows a source file that contains lines too long to be handled by the destination device, for example, an 80-character printer or narrow console. The number n should be the maximum column width of the destination device.
E	Echo transfer at console. When this parameter follows a source name, PIP displays the source data at the console as the copy is taking place. The source must contain character data.
F	Filter form feeds. When this parameter follows a source name, PIP removes all form feeds embedded in the source data. To change form feeds set for one page length in the source file to another page length in the destination file, use the F command to delete the old form feeds and a P command to simultaneously add new form feeds to the destination file.
Gn	Get source from, or go to user number n. When this parameter follows a source name, PIP searches the directory of user number n for the source file. When it follows the destination name, PIP places the destination file in the user number specified by n. The number must be in the range 0 to 15.

Table 5-2. (continued)

Option	Function
H	Hex data transfer. PIP checks all data for proper Intel® hexadecimal file format. The console displays error messages when errors occur. This function is included for historical reasons only.
I	Ignore :00 records in the transfer of an Intel hexadecimal format file. The I option automatically sets the H option. This function is included for historical reasons only.
L	Translate uppercase alphabetics in the source file to lowercase in the destination file. This parameter follows the source device or filename.
N	Add line numbers to the destination file. When this parameter follows the source filename, PIP adds a line number to each line copied, starting with 1 and incrementing by one. A colon follows the line number. If N2 is specified, PIP adds leading zeros to the line number and inserts a tab after the number. If the T parameter is also set, PIP expands the tab.
O	Object file transfer for machine code (noncharacter and therefore nonprintable) files. PIP ignores any CTRL-Z end-of-file during concatenation and transfer. Use this option if you are combining object code files.

Table 5-2. (continued)

Option	Function
Pn	Set page length. n specifies the number of lines per page. When this parameter modifies a source file, PIP includes a page eject at the beginning of the destination file and at every n lines. If n=1 or is not specified, PIP inserts page ejects every 60 lines. When you also specify the F option, PIP ignores form feeds in the source data and inserts new form feeds in the destination data at the page length specified by n.
Qs	Quit copying from the source device after the string s. When used with the S parameter, this parameter can extract a portion of a source file. The string argument must be terminated by CTRL-Z.
R	Read system (SYS) files. Usually, PIP ignores files marked with the system attribute in the disk directory, but when this parameter follows a source filename, PIP copies system files, including their attributes, to the destination.
Ss	Start copying from the source device at the string s. The string argument must be terminated by CTRL-Z. When used with the Q parameter, this parameter can extract a portion of a source file. Both start and quit strings are included in the destination file.
Tn	Expand tabs. When this parameter follows a source filename, PIP expands tab (CTRL-I) characters in the destination file. PIP replaces each CTRL-I with enough spaces to position the next character in a column divisible by n.

Table 5-2. (continued)

Option	Function
U	Translate lowercase alphabetic characters in the source file to uppercase in the destination file. This parameter follows the source device or filename.
V	Verify that data has been copied correctly. PIP compares the destination data to the source data to ensure that the data has been written correctly. The destination must be a disk file.
W	Write over files with RO (Read-Only) attribute. Usually, if a PIP command tail includes an existing RO file as a destination, PIP sends a query to the console to make sure you want to write over the existing file. When this parameter follows a source name, PIP overwrites the RO file without a console exchange. If the command tail contains multiple source files, this parameter need only follow the last file in the list.
Z	Zero, the parity bit. When this parameter follows a source name, PIP sets the parity bit of each data byte in the destination file to zero. The source must contain character data.

Examples: The following command constructs the file NEWPROG.BAS on drive A by joining the two files CODE.BAS and DATA.BAS from drive A. During the copy operation, CODE.BAS is translated to lowercase, while DATA.BAS is translated to uppercase.

```
A>PIP NEWPROG.BAS=CODE.BAS[L], DATA.BAS[U]
```

The next command writes the character file WIDEFILE.SRC from drive A to the console device, but deletes all characters following the 80th column position.

```
A>PIP CON:=WIDEFILE.SRC[D80]
```

In the next example, the file LETTER.TXT from drive A is copied to LETTER.TXT on drive B. The LETTER.TXT file is also written to the screen as the copy operation proceeds.

```
A>PIP B:=LETTER.TXT[E]
```

The next command writes the file LONGPAGE.TXT from drive B to the printer device. As the file is written, form-feed characters are removed and reinserted at the beginning and every 65th line thereafter.

```
A>PIP LST:=B:LONGPAGE.TXT[FP65]
```

The next command writes the file PROGRAM.ASM from drive B to the printer device. The N parameter tells PIP to number each line. The T8 parameter expands tabs to every eighth column. The U parameter translates lowercase letters to uppercase as the file is printed.

```
B>PIP LST:=PROGRAM.ASM[NT8U]
```

The following command extracts a portion of the LETTER.TXT file from drive A by searching for the character sequence "Dear Sir" before starting the copy operation. When found, the characters are copied to PORTION.TXT on drive A until the sequence "Sincerely" is found in the source file. Note that this PIP option can only be used when PIP is in the multiple copy mode with the asterisk prompt, even if you do not intend to make multiple copies. This option cannot be used in single copy mode on the same line as the PIP command keyword.

```
A>PIP
*PORTION.TXT=LETTER.TXT[SDear Sir^Z QSincerely^Z]
```

The next command copies all files with filetype CMD from drive A to drive B. The V parameter tells PIP to read the destination files to ensure that data was correctly transferred. The W parameter lets PIP overwrite any destination files that are marked as RO (Read-Only). The R parameter tells PIP to read files from drive A that are marked with the SYS (System) attribute.

```
B>PIP B:=A:*.CMD[VWR]
```

REN Command

Syntax: REN [new-filespec=old-filespec]

Explanation: The REN command lets you change the name of a file that is cataloged in the directory of a disk.

The new-filespec must not be the name of any existing file on the disk. The old-filespec identifies an existing file on the disk.

The REN command changes the name of the file named by old-filespec to the name given as new-filespec. If you put a space between the old-filespec and the equal sign or between the new-filespec and the equal sign, you must insert at least one space on the opposite side of the equal sign.

CP/M-8000 offers you a second method of using the REN command. Type REN then press RETURN. REN presents the following display:

```
Enter Old Name:_
```

The cursor appears following Enter Old Name:. Enter the filespec you want to rename and press RETURN. The following line appears on your screen:

```
Enter New Name:_
```

The cursor then appears following Enter New Name:. Type in the new-filespec and press RETURN to execute the REN command.

REN does not make a copy of the file. REN changes only the name of the file.

If you omit the drive specifier, REN assumes the file to rename is on the default drive. You can include a drive specifier as a part of the new name. If both file specifications name a drive, it must be the same drive.

If the file given by the old name does not exist, REN displays the following message on the screen:

```
No File
```

If the file given by the new name is already present in the directory, REN displays the following message on the screen:

File Already Exists

REN does not allow for wildcard filenames or filetypes. If you try to use a wildcard, REN tells you:

No wildcard filenames

Examples: The following command renames the file MORTAL.LST to IMMORTAL.LST on drive A.

```
A>REN IMMORTAL.LST=MORTAL.LST
```

If you enter the following command:

```
A>REN
```

The system prompts for the filespecs:

```
Enter Old Name:X.PRN

Enter New Name:Y.PRN

A>
```

File X.PRN is renamed Y.PRN on drive A.

In the following example, the file Y.PLI changes to X.PAS on drive A.

```
B>REN A:X.PAS=Y.PLI
```

In the next command, the file OLDLIST changes to NEWLIST on drive B. Because the second drive specifier, B: is implied by the first one, it is unnecessary in this example.

```
A>REN B:NEWLIST=B:OLDLIST
```

The command line above has the same effect as the following:

```
A>REN B:NEWLIST=OLDLIST
```

or

```
A>REN NEWLIST=B:OLDLIST
```

STAT Command

Syntax:

```
STAT
STAT filespec [RO|RW|SYS|DIR|SIZE]
STAT [d:]DSK: | USR:
STAT VAL: | DEV:
```

Explanation: The various forms of the STAT command give you information about the disk drives, files, and devices associated with your computer. STAT lets you change the attributes of files. You can also assign physical devices to the STAT logical device names.

Note that the options following filespec can appear in square brackets, [], or without delimiters as shown in the syntax section above.

The notation RW tells you that a file is in a Read-Write state so that data can be both read from and written to the file. A file marked RO, Read-Only, can be read from but not written to.

Unlike other CP/M products, you cannot use STAT in CP/M-8000 to set a drive to a Read-Only state. Drives are in a Read-Write state by default and become Read-Only only when you remove a disk on which a file is open. The drive reverts to a Read-Write state whenever you exit a program or type a CTRL-C, either of which causes a warm boot. See Section 4 for a discussion of terminating programs and the term "warm boot."

Except for STAT VAL: and STAT DEV:, the STAT commands allow you to use a drive specifier. In CP/M-8000, you can specify a drive that is different from the currently logged-in drive. The drive letter that appears in your system prompt identifies the logged-in drive. Execution of a STAT command with a drive specifier that differs from the logged-in drive does not change the logged-in drive.

In a STAT command, when you specify a drive that is different from the logged-in drive, you place that drive in active status. Active status is any drive accessed since the last warm or cold start. If you execute a STAT or STAT DSK: command without specifying a drive, STAT responds with information for any or all drives on active status.

Free Space on Disk

Syntax: STAT [d:]

Explanation: STAT with no command tail reports the amount of free storage space that is available on all on-line disks. This form of the STAT command reports free space for only those disks that have been accessed since CP/M-8000 was last started or reloaded. You can find the amount of free space on a particular disk by including the drive specifier in the command tail.

This form of the STAT command displays information on your screen in the following form:

d: RW, Free Space: nnK

where d is the drive specifier, and n is the number of kilobytes of storage remaining on the disk in the drive specified by d.

Examples: Suppose you have two disk drives containing active disks. Suppose also that drive A has 16K (16,384) bytes of free space, while drive B has 32K (32,768) bytes of free space. After you enter the STAT command, you see the following messages display on your screen:

```
A>STAT

A: RW, Free Space:  16k
B: RW, Free Space:  32k
```

Suppose drive B has 98 kilobytes of storage that is free for program and data storage. Enter the STAT command, specifying drive B; the following message is displayed on your screen:

```
A>STAT B:

B: RW, Free Space: 98K
```

Files - Display File Space and Access Mode

Syntax: STAT filespec [SIZE]

Explanation: This form of the STAT command displays the kilobytes of space used by the specified file. It also displays the Access Mode of the file. STAT accepts wildcards in the filename and filetype part of the command tail. When you include a wildcard in your file specification, the STAT command displays a list of qualifying files, with their file characteristics, from the default or specified drive. STAT also sorts this list alphabetically.

Note that the SIZE option following the filespec can be enclosed in square brackets, [], be preceded by a dollar sign, $, or be included in the command with no delimiter, as in the preceding syntax line.

CP/M-8000 supports four file Access Modes:

RO The file has the Read-Only attribute that allows data to come from the file, but the file cannot be altered.

RW The file has the Read-Write attribute that allows data to move either to or from the file.

SYS The file has the system attribute. System files do not appear in DIR (directory) displays. Use DIRS to show system (SYS) files. Use the STAT command to display all files including those with the system attribute. The STAT command shows system files in parentheses.

DIR The file has the directory attribute and appears in DIR (directory) displays.

A file has either the RO or RW attribute, and either the SYS or DIR attribute. By default, and unless changed by the STAT command, a file has the RW and DIR attributes.

This format for the STAT command produces a list of file characteristics under five headings:

- The first column displays the number of records used by the file, where each record is 128 bytes in length. This value is listed on your screen under the column marked Recs.

- The second column displays the number of kilobytes used by the file, where each kilobyte contains 1,024 bytes. This value is listed under Bytes.

- The third column displays the number of directory entries used by the file. This value appears under the FCBs column. FCB (File Control Block) is another name for a directory entry.

- The Access Modes are displayed under the Attributes column.

- The file specification, consisting of the drive specifier, filename, and filetype of the file appears under Name on your screen.

Use the SIZE option to tell STAT to compute the virtual file size of each file. The virtual and real file size are identical for sequential files, but can differ for files written in random mode. When you use SIZE, the additional column, marked Size, is displayed on the screen. The value in this column represents the number of filled and unfilled records allotted to the file.

When you enter the command STAT *.*, STAT performs a directory verification to ensure that two files do not share the same disk space allocation, that is, the indicated file dies not share a portion of the disk with another file in the directory. If STAT finds a duplicate space allocation it displays the following message:

```
Bad Directory on d:
Space Allocation Conflict:
User nn  d:filename.typ
```

STAT prints the user number and the name of the file containing doubly allocated space. More than one file can be listed. The recommended solution is to erase the listed files, and then type a CTRL-C.

STAT does a complete directory verification whenever a wildcard character appears in the command tail.

Examples: The following command tells STAT to display the characteristics of all files on drive B that begin with the letters MY, and have any filetype at all.

```
A>STAT B:MY*.*
```

Assume that drive B contains three files that satisfy the file specification. The screen could display the following:

```
Drive B:                                          User 0
Recs   Bytes   FCBs   Attributes                 Name
  16      2K      1   Dir RW                  B:MYPROG   .8KN
   8      1K      1   Dir RO                  B:MYTEST   .DAT
  32     18K      2   Sys RO                  B:MYTRAN   .Z8K
.......................................................
Total: 23K      4     (3 files,        21 - 1K blocks)
B: RW, Free Space:          243K
```

The next command causes the same action as the previous command, but includes the Size column in the display.

```
A>STAT B:MY*.* SIZE
```

Assume that MYTEST.DAT was written using random access from record number 8 through 15, leaving the first 8 records empty. The virtual file size is 16 records, although the file only consumes eight records. The screen appears as follows:

```
Drive B:                                          User 0
Size  Recs  Bytes  FCBs  Attributes               Name
  16    16     2K     1  Dir RW                 B:MYPROG   .8KN
  16     8     1K     1  Dir RO                 B:MYTEST   .DAT
  32    32    18K     2  Sys RO                 B:MYTRAN   .Z8K
.......................................................
Total:        23K     4  (5 files,       21 - 1K blocks)
B: RW, Free Space:          243K
```

Note that the figure for total bytes, 23K, includes directory space as well as space used for files.

Set File Access Modes (Attributes)

Syntax: STAT filespec RO |RW |SYS |DIR

Explanation: This form of the STAT command lets you set the Access Mode for one or more files. Note that the option following filespec can be enclosed in square brackets, [], be preceded by a dollar sign, $, or be included with no delimiter as shown above.

The four Access Modes, described above, are:

RO
RW
SYS
DIR

A file can have either the RO or RW Access Mode, but not both. Similarly, a file can have either the SYS or DIR Access Mode, but not both.

Examples: This command sets the Access Mode for the file LETTER.TXT on the default drive to Read-Only.

```
A>STAT LETTER.TXT RO
```

The following message appears on your screen if the file is present:

```
LETTER.TXT set to RO
```

The following command:

```
B>STAT A:*.Z8K SYS
```

sets the Access Mode for all files on drive A with filetype Z8K to SYS. Given that the three command files PIP, ED, and STAT are present on drive A, the following message appears on your screen:

```
PIP.Z8K set to SYS
ED.Z8K set to SYS
STAT.Z8K set to SYS
```

Display Disk Status

Syntax: STAT [d:]DSK:

Explanation: This form of the STAT command displays information concerning the internal characteristics of your disk storage system for all on-line disk drives placed in active status. This form of the STAT command is provided for advanced programming functions, and is not necessary for your everyday use of CP/M-8000.

Examples: This STAT command displays information about drive A and any other drives that have been placed in active status:

```
A>STAT DSK:
```

STAT supplies numbers for n in the following form:

```
   A: Drive Characteristics
nnnn: 128 Byte Record Capacity
nnnn: Kilobyte Drive  Capacity
nnnn: 32 Byte Directory Entries
nnnn: Checked  Directory Entries
nnnn: 128 Byte Records/Directory Entry
nnnn: 128 Byte Records/Block
nnnn: 128 Byte Records/Track
nnnn: Reserved Tracks
```

The following command produces the type of information shown in the previous example, but for drive B.

```
A>STAT B:DSK:
```

Display User Numbers with Active Files

Syntax: STAT [d:]USR:

Explanation: This form of the STAT command lets you determine the user numbers that have files on the disk in the specified drive.

User numbers are assigned to files that are created under CP/M-8000. Use this form of the STAT command to determine the active user numbers on a disk.

Examples: This command displays the user numbers containing active files on the disk in drive A.

```
A>STAT USR:
```

Display STAT Commands and Device Names

Syntax: STAT VAL:

Explanation: STAT VAL: displays a summary of the general forms of the STAT commands. It also displays the possible physical device names that you can assign to each of the four CP/M-8000 logical device names.

Examples: The STAT VAL: display is shown below:

```
A>STAT VAL:

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>
```

Display and Set Physical-to-Logical Device Assignments

Syntax:
```
STAT DEV:
STAT logical device: = physical device:
```

Explanation: STAT DEV: displays the current assignments for the four CP/M-8000 logical device names: CON:, RDR:, PUN:, and LST:. Use the second form of the above STAT command to change these current assignments. The command STAT VAL: displays the possible physical device names that you can assign to each logical device name. Refer to the part of the STAT VAL: display entitled, Iobyte Assign, shown above.

When you assign a physical device to a logical device, STAT assigns a value from 0 to 3 to the logical device name in what is called the Iobyte.

You can assign any of the listed physical device names to their appropriate logical device names. However, the assignment does not work unless you are using the proper Input/Output Port (I/O) on your computer, with the proper cable to connect the computer to the device, and the proper I/O driver routine for the particular physical device. This facility must be supported by the manufacturer of your computer before the physical-to-logical assignments can be meaningful.

Examples: **A>STAT CON: = CRT:**

This command assigns the physical device name CRT: to the logical input device name CON:, which generally refers to the console.

A>STAT LST: = LPT:

The command above assigns the physical device name LPT: to the logical output device name LST:, which generally refers to the list device or the printer.

SUBMIT Command

Syntax: [SUBMIT] [filespec (argument1 ... argument9)]

Explanation: The SUBMIT command lets you execute a group or batch of commands from a SUB file, which is a file with a filetype of SUB.

Usually, you enter commands one line at a time. If you must enter the same sequence of commands several times, you might find it easier to batch the commands together using the SUBMIT command. To do this, you create a file and enter your commands in this file. The file is identified by the filename, and must have a filetype of SUB. When you issue the SUBMIT command, SUBMIT reads the file named by the filespec and prepares it for interpretation by CP/M-8000. When the preparation is complete, SUBMIT sends the file to CP/M-8000 line by line, as if you were typing each command.

Example: The following sample lines illustrate some of the variety of commands that can be entered in a SUB file:

```
;This is an example SUBMIT file comment.
DIR
DIR *.BAK
ERA *.BAK
PIP LST:=$1.PRN[T$2 $3 $4]
DIRS *.Z8K
PIP A:*.LTR
DIR B:
```

Creating the SUB File

Explanation: The SUB file can contain any valid CP/M-8000 command or any valid CP/M-8000 command with SUBMIT parameters. You can even reference other SUB files in a SUB file. Embedded files of this kind are treated as chained files, not nested files. That is, any commands following an embedded SUB file will be ignored; a SUB file called by another SUB file never returns to the caller after successful execution. For that reason, the last line in a SUB file is the only useful place for an embedded SUB file.

A semicolon is the delimiter used for a comment in a SUBMIT file. The comment must be entered on a line by itself and start in the first column of that line. Comments print on the console exactly as they are in the SUBMIT file.

You can put more than one command on a line if you separate each command with an exclamation mark. For example,

```
A>ERA *.BAK ! DIR *.Z8K ! PIP $1:=$2:$3[V]
```

contains three distinct CP/M-8000 commands on one line of a SUBMIT file. The length of the line must not exceed 128 characters.

You can create the SUB file with the ED utility, with some other editor, or with a word processing program.

You can pass arguments to SUB files when you execute them. Each argument you enter is assigned to a parameter in the SUB file. The first argument replaces every occurrence of $1 in the file, the second argument replaces parameter $2, etc., up to parameter $9. The $0 parameter always substitutes for the SUBMIT file itself. For example, if your file START.SUB contains the following commands:

```
ERA $1.BAK
DIR $1
PIP A:$1=A:$2.Z8K
PIP A:FINISH.SUB=A:$0.SUB
```

and you enter the following SUBMIT command:

```
A>SUBMIT START SAM TEX
```

SUBMIT substitutes these actual parameters for the symbolic stand-in parameters in your START.SUB file as sequential execution occurs. In this example, the argument SAM is substituted for every $1 in the START.SUB file as sequential execution occurs, TEX is substituted for every occurrence of $2 in the START.SUB file, and START is substituted for $0. As each line is encountered in your SUB file, SUBMIT then executes this file, line by line. The substitutions look like this to SUBMIT:

```
ERA SAM.BAK
DIR SAM
PIP A:SAM=A:TEX.Z8K
PIP A:FINISH.SUB=START.SUB
```

If you enter fewer arguments in the SUBMIT command than there are parameters in the SUB file, the extra parameters are not used.

If you enter more arguments in the SUBMIT command than there are parameters in the SUB file, the extra arguments are ignored.

To include an actual dollar sign, $, in your SUB file, type two dollar signs in a row, $$. SUBMIT replaces them with a single dollar sign. For example, if you have a file SEARCH.SUB and it contains the line:

```
DIR $1:*.$$$$$$
```

and you enter the following SUBMIT command,

```
A>SUBMIT SEARCH B
```

then the translated file contains the following:

```
DIR B:*.$$$
```

The net effect in this example is that CP/M-8000 looks for all files on drive B that have the filetype of $$$.

Executing the SUBMIT Command

Syntax: To appreciate how SUBMIT is executed, recall the syntax mentioned earlier:

{SUBMIT} {filespec (argument1 ... argument9)}

Explanation: Based on the above syntax, there are five ways of executing a SUBMIT command before the parameter list is evaluated:

```
SUBMIT filename.SUB
SUBMIT filename
filename.SUB
filename
SUBMIT
```

If the first form is used, CP/M-8000 first looks in your current user number for the filename you typed with the filetype of SUB. If that file specification is found there, SUBMIT tries to execute the file. If it is not found in your current user number, CP/M-8000 then searches user 0 for filename.SUB. If it is not found in either user number, CP/M-8000 displays your filename followed by a question mark on your console.

If the second form is used, CP/M-8000 appends the SUB filetype to your filename and searches for that file specification in your current user number, then in user 0, if need be. If that file specification is found, SUBMIT tries to execute the file. If it is not found, CP/M-8000 displays your filename followed by a question mark on your console.

If the third form is used, CP/M-8000 assumes you want to invoke the SUBMIT command because you supplied the SUB filetype. If that file specification is found in your current user number (or user 0), SUBMIT tries to execute the file. If it is not found, CP/M-8000 displays your filename followed by a question mark on your console.

If the fourth form is used, the general rules of file searching as set forth in Section 4 take precedence as shown in the following four steps:

1. CP/M-8000 assumes you are trying to invoke a program file in your current user number, but CP/M-8000 does not assume you supplied a filetype. Therefore, CP/M-8000 first appends a filetype of Z8K to your filename and searches your current user number for the file. If a file with your filename and a Z8K filetype is found, CP/M-8000 executes that file without invoking SUBMIT.

2. If a file with a Z8K filetype is not found, CP/M-8000 substitutes a blank filetype for the Z8K filetype it appended to your filename and searches for this file specification in your current user number. If such a file exists, it is executed and SUBMIT is again bypassed.

3. If the file specification with the blank filetype is not found, CP/M-8000 substitutes a SUB filetype for the blank filetype and searches for this file specification in your current user number. If filename.SUB is found, CP/M-8000 recognizes that you want to execute a SUBMIT file; control passes to SUBMIT, and SUBMIT tries to execute your file.

4. If CP/M-8000 is unable to find any of the above combinations in your current user number, it searches user 0 first for your filename.Z8K, then for your filename.<blank>, and finally for your filename.SUB. If none of these three file specifications are found in either the current user number or user 0, CP/M-8000 displays the command keyword followed by a question mark.

If the fifth form (SUBMIT) is used, CP/M-8000 prompts you for your file specification:

Enter Filename: _

The underscore represents your cursor's position. You can then reply with a filename having a blank filetype or with a filename having a filetype of SUB. If you provide a blank filetype, CP/M-8000 appends the SUB filetype for you. If you provide any filetype other than SUB or blank, CP/M-8000 rejects it as inappropriate for SUBMIT and asks you to supply a filename with a filetype of SUB or blank.

Example: Enter SUBMIT at the system prompt:

A>**SUBMIT**

the system prompts:

Enter Filename:

Enter the filespec and arguments here, such as

START B TEX

Another example could be:

A>**SUBMIT BUBBA**

where you enter the filename, BUBBA, after the program name, SUBMIT, on the command line.

TYPE Command

Syntax: TYPE [filespec]

Explanation: The TYPE built-in command displays the contents of a character file on your screen. If the file occupies more than a screenful of space, TYPE scrolls the file vertically up the screen. Press CTRL-S to stop the scrolling and CTRL-Q to resume. Wildcards are not permitted with this command.

Tab characters occurring in the file named by the file specification are expanded to every eighth column position of your screen.

To abort the TYPE command while text is scrolling on your screen, press CTRL-C.

Be sure that the file specification you enter identifies a file containing character data.

Examples: The following command displays the contents of the file MYPROG.ASM on your screen.

```
A>TYPE MYPROG.ASM
```

The next command displays the contents of the file THISFILE from drive B on your screen.

```
A>TYPE B:THISFILE
```

TYPE can also be executed without a file specification in the command line. When you use TYPE in this manner, you are prompted for your file specification:

```
Enter Filename: _
```

The underscore represents your cursor's position. Respond with the file specification of the file whose contents you want to see displayed on your screen.

If the file named by the file specification is not present on an on-line disk, TYPE displays the following message on your screen:

```
No File
```

To list the file at the printer as well as on the screen, type a CTRL-P before entering the TYPE command line. To stop echoing keyboard input at the printer, type a second CTRL-P.

USER Command

Syntax: USER [number]

Explanation: USER is a built-in command used to set the current user number. The disk directory can be divided into distinct groups according to user numbers ranging from 0 through 15.

When CP/M-8000 starts, 0 is the current user number. Executable files created under user 0 are accessible from any other user number. Files that you create under any user number other than 0 are not accessible under any other user number, except through the PIP command. See the G option of the PIP Command.

Example: The following command changes the current user number from 0 to 3. Note that the prompt changes to 3A>.

```
A>USER 3
3A>
```

USER can also be executed without specifying the user number in the command line. If you do this, USER prompts you for your user new number.

```
A>USER

Enter User No: _
```

The underscore represents your cursor's position. Respond with the user number you want to make current.

```
Enter User No:5
5A>
```

The current user number is now 5 on drive A.

End of Section 5

CP/M-8000 Editor

Introduction to ED

To do almost anything with a computer you need some way to enter data, some way to give the computer the information you want it to process. The programs most commonly used for this task are called editors. They transfer your keystrokes at the keyboard to a disk file. CP/M-8000's editor is named ED. Using ED, you can create and alter CP/M-8000 text files.

The correct command syntax for invoking the CP/M-8000 editor is given in "Starting ED." After starting ED, you issue commands that transfer text from a disk file to memory for editing. "ED Operation" details this operation and describes the basic text transfer commands that allow you to easily enter and exit the editor.

"Basic Editing Commands" details the commands that edit a file. "Combining ED Commands" describes how to combine the basic commands to edit more efficiently. Although you can edit any character file with the basic ED commands, ED provides several more commands that perform more complicated editing functions, as described in "Advanced ED Commands."

During an editing session, ED can return two types of error messages. "ED Error Messages" lists these messages and provides examples that indicate how to recover from common editing error conditions.

Starting ED

Syntax: ED filespec filespec

To start ED, enter its name after the CP/M-8000 prompt, followed by a file specification that contains no wildcard characters. The file specification, MYFILE.TEX in the following example, specifies a file to be edited or created. The file specification can be preceded by a drive specifier but a drive specifier is unnecessary if the file to be edited is on your default drive.

```
A>ED MYFILE.TEX
```

Optionally, the file specification can be followed by a drive specifier, as shown in the following example.

```
A>ED MYFILE.TEX B:
```

In response to this command, ED opens the file to be edited, MYFILE.TEX, on drive A, but sends all the edited material to a file on drive B.

Optionally, you can send the edited material to a file with a different filename, as shown in the following example.

```
A>ED MYFILE.TEX YOURFILE.TEX
```

The file with the different filename cannot already exist or ED prints the following message and terminates.

```
Output File Exists, Erase It
```

The ED prompt, *, appears at the screen when ED is ready to accept a command, as shown below.

```
A>ED MYFILE.TEX
     : *
```

If no previous version of the file exists on the current disk, ED automatically creates a new file and displays the following message:

```
NEW FILE
     : *
```

Note: 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. It should be larger if you plan to add characters to the file. When ED finds a disk or directory full, ED has only limited recovery mechanisms. These are explained in "ED Error Messages."

ED Operation

With ED, you change portions of a file that pass through a memory buffer. When you start ED with one of the commands shown above, this memory buffer is empty. At your command, ED reads segments of the source file, for example MYFILE.TEX, into the memory buffer for you to edit. If the file is new, you must insert text into the file before you can edit. During the edit, ED writes the edited text onto a temporary work file, MYFILE.$$$.

When you end the edit, ED writes the memory buffer contents to the temporary file, followed by any remaining text in the source file. ED then changes the name of the source file from MYFILE.TEX to MYFILE.BAK, so you can reclaim this original material from the backup file if necessary. ED then renames the temporary file, MYFILE.$$$, to MYFILE.TEX, the new edited file. The following figure illustrates the relationship between the source file, the temporary work file and the new file.

Note: When you invoke ED with two filespecs, an input and output file, ED does not rename the input file to type .BAK; therefore, the input file can be Read-Only or on a write-protected disk if the output file is written to another disk.

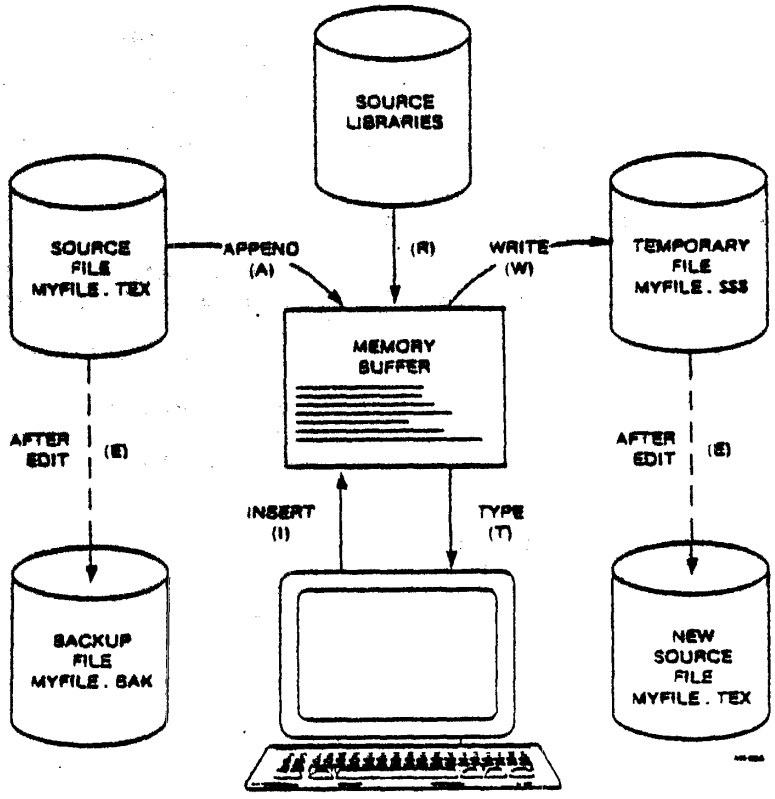


Figure 6-1. Overall ED Operation

In Figure 6-1, the memory buffer is logically between the source file and the temporary work file. ED supports several commands that transfer lines of text between the source file,

the memory buffer, and the temporary (and eventually final) file. The following table lists the three basic text transfer commands that allow you to easily enter the editor, write text to the temporary file, and exit the editor.

Table 6-1. Text Transfer Commands

Command	Result
nA	Append the next n unprocessed source lines from the source file to the end of the memory buffer.
nW	Write the first n lines of the memory buffer to the temporary file free space.
E	End the edit. Copy all buffered text to the temporary file, and copy all unprocessed source lines to the temporary file. Rename files.

Appending Text into the Buffer

When you start ED and the memory buffer is empty, you can use the A (append) command to add text to the memory buffer.

Note: ED can number lines of text to help you keep track of data in the memory buffer. The colon that appears when you start ED indicates that line numbering is turned on. Type -V after the ED prompt to turn the line number display off. Line numbers appear on the screen but never become a part of the output file.

A (Append) Command

The A command appends (copies) lines from an existing source file into the memory buffer; it takes the form:

nA

where n is the number of unprocessed source lines to append into the memory buffer. If a pound sign, #, is given in place of n, then the integer 65535 is assumed. Because the memory buffer can contain most reasonably sized source files, it is often possible to issue the command #A at the beginning of the edit to read the entire source file into memory.

If n is 0, ED appends the unprocessed source lines into the memory buffer until the buffer is approximately half full. If you do not specify n, ED appends one line from the source file into the memory buffer.

ED Exit

You can use the W (Write) command or the E (Exit) command to save your editing changes. The W command writes lines from the memory buffer to the new file without ending the ED session. An E command saves the contents of the buffer and any unprocessed material from the source file and exits ED.

W (Write) Command

The W command writes lines from the buffer to the new file; its form is:

nW

where n is the number of lines to be written from the beginning of the buffer to the end of the new file. If n is greater than 0, ED writes n lines from the beginning of the buffer to the end of the new file. If n is 0, ED writes lines until the buffer is half empty. The 0W command is a convenient way of making room in the memory buffer for more lines from the source file. You can determine the number of lines to write out by executing a 0V command to check the amount of free space in the buffer, as shown below:

```
1: *0V
25000/30000
1: *
```

The preceding display indicates that the total size of the memory buffer is 30,000 bytes and there are 25,000 free bytes in the memory buffer.

Note: After a W command is executed, you must enter the H command to reedit the saved lines during the current editing session.

E (Exit) Command

An E command performs a normal exit from ED. The form of the E command is:

E

followed by a carriage return.

When you enter an E command, ED first writes all data lines from the buffer and the original source file to the new file. If a .BAK file exists, ED deletes it, then renames the original file with the .BAK filetype. Finally, ED renames the new file from filename.$$$ to the original filetype and returns control to the operating system.

The operation of the E command makes it unwise to edit a backup file. When you edit a BAK file and exit with an E command, ED erases your original file because it has a .BAK filetype. To avoid this, always rename a backup file to some other filetype before editing it with ED.

Note: Any command that terminates an ED session must be the only command on the line.

Basic Editing Commands

The text transfer commands discussed above allow you to easily enter and exit the editor. This section discusses the basic commands that edit a file.

ED treats a file as a long chain of characters grouped together in lines. ED displays and edits characters and lines in relation to an imaginary device called the character pointer (CP). During an edit session, you must mentally picture the CP's location in the memory buffer and issue commands to move the CP and edit the file.

The following commands move the character pointer or display text in the vicinity of the CP. These ED commands consist of a numeric argument and a single command letter and must be followed by a carriage return. The numeric argument, n, determines the number of times ED executes a command; however, there are four special cases to consider in regard to the numeric argument:

- If the numeric argument is omitted, ED assumes an argument of 1.
- Use a negative number if the command is to executed backwards through the memory buffer. (The B command is an exception).
- If you enter a pound sign, #, in place of a number, ED uses the value 65535 as the argument. A pound sign argument can be preceded by a minus sign (-#) to cause the command to execute backwards through the memory buffer.
- ED accepts 0 as a numeric argument only in certain commands. In some cases, 0 causes the command to be executed approximately half the possible number of times, while in other cases it prevents the movement of the CP.

The following table alphabetically summarizes the basic editing commands and their valid arguments.

Table 6-2. Basic Editing Commands

Command	Action
B, -B	Move CP to the beginning (B) or end (-B) of the memory buffer.
nC, -nC	Move CP n characters forward (nC) or backward (-nC) through the memory buffer.
nD, -nD	Delete n characters before (-nD) or after (nD) the CP.
I	Enter insert mode.
Istring↑Z	Insert a string of characters.
nK, -nK	Delete (kill) n lines before the CP (-nK) or after the CP (nK).
nL, -nL	Move the CP n lines forward (nL) or backward (-nL) through the memory buffer.
nT, -nT	Type n lines before the CP (-nT) or after the CP (nT).
n, -n	Move the CP n lines before the CP (-n) or after the CP (n) and display the destination line.

The following sections discuss ED's basic editing commands in more detail. The examples in these sections illustrate how the commands effect the position of the character pointer in the memory buffer. Later examples in "Combining ED Commands" illustrate how the commands appear at the screen. For these sections, however, the circumflex symbol (^) in command examples represents the character pointer, which you must imagine in the memory buffer.

Moving the Character Pointer

This section describes commands that move the character pointer in useful increments but do not display the destination line. Although ED is used primarily to create and edit program source files, the following sections present a simple text as an example to make ED easier to learn and understand.

B (Beginning/Bottom) Command

The B command moves the CP to the beginning or bottom of the memory buffer. The B command has two forms:

B, -B

-B moves the CP to the end or bottom of the memory buffer; B moves the CP to the beginning of the buffer.

C (Character) Command

The C command moves the CP forward or backward the specified number of characters; it takes the form:

nC, -nC

where n is the number of characters the CP is to be moved. A positive number moves the CP towards the end of the line and the bottom of the buffer. A negative number moves the CP towards the beginning of the line and the top of the buffer. You can enter an n large enough to move the CP to a different line. However, each line is separated from the next by two invisible characters: a carriage-return and a line feed represented by <cr><lf>. You must compensate for their presence. For example, if the CP were positioned before the E of Emily, the command 30C moves the CP to the next line:

```
Emily Dickinson said,<cr><lf>
"I fin^d ecstasy in living -<cr><lf>
```

L (Line) Command

The L command moves the CP the specified number of lines. After an L command, the CP always points to the beginning of a line. The forms of the L command are:

nL, -nL

where n is the number of lines the CP is to be moved. A positive number moves the CP towards the end of the buffer. A negative number moves the CP back toward the beginning of the buffer. The command 2L moves the CP from the first example line two lines forward through the memory buffer and positions the character pointer at the beginning of the line.

```
 Emily Dickinson said,<cr><lf>
 "I find ecstasy in living -<cr><lf>
^the mere sense of living<cr><lf>
```

The command -L moves the CP to the beginning of the previous line, even if the CP originally points to a character in the middle of the line. Use the special character 0 to move the CP to the beginning of the current line.

n (Number) Command

The n command moves the CP and displays the destination line. The forms of the n command are

n, -n

where n is the number of lines the CP is to be moved. In response to this command, ED moves the CP forward or backward the number of lines specified, then prints only the destination line.

```
 Emily Dickinson said,<cr><lf>
^"I find ecstasy in living -<cr><lf>
```

A further abbreviation of this command is to enter no number at all. In response to a carriage return without a preceding command, ED assumes an n command of 1 and moves the CP down to the next line and prints it.

```
 Emily Dickinson said,<cr><lf>
^"I find ecstasy in living -<cr><lf>
```

Also, a minus sign, -, without a number moves the CP back one line.

Displaying Memory Buffer Contents

ED does not display the contents of the memory buffer until you specify which part of the text you want to see. The T command displays text without moving the CP.

T (Type) Command

The T command types a specified number of lines from the CP to the screen. The T command takes the forms:

nT, -nT

where n specifies the number of lines to be displayed. If a negative number is entered, ED displays n lines before the CP. A positive number displays n lines after the CP. If no number is specified, ED types from the character pointer to the end of the line. The CP remains in its original position no matter how many lines are typed. For example, if the character pointer is at the beginning of the memory buffer, and you instruct ED to type four lines (4T), four lines are displayed at the screen, but the CP stays at the beginning of line 1.

```
^Emily Dickinson said,<cr><lf>
 "I find ecstasy in living -<cr><lf>
 the mere sense of living
 is joy enough."
```

If the CP is between two characters in the middle of the line, a T command with no number specified types only the characters between the CP and the end of the line, but the character pointer stays in the same position, as shown in the memory buffer example below.

```
"I find ec^stasy in living -
```

Whenever ED is displaying text with the T command, you can enter a CTRL-S to stop the display, then a CTRL-Q when you're ready to continue scrolling. Enter a CTRL-C to abort long type-outs.

Deleting Characters

D (Delete) Command

The D command deletes a specified number of characters and has the forms:

```
nD, -nD
```

where n is the number of characters to be deleted. If no number is specified, ED deletes the character to the right of the CP. A positive number deletes multiple characters to the right of the CP, towards the bottom of the file. A negative number deletes characters to the left of the CP, towards the top of the file. If the character pointer is positioned in the memory buffer as shown below,

```
Emily Dickinson said,<cr><lf>
"I find ecstasy in living -<cr><lf>
the mere sense of living<cr><lf>
is joy ^enough."<cr><lf>
```

the command 6D deletes the six characters after the CP, and the resulting memory buffer looks like this:

```
Emily Dickinson said,<cr><lf>
"I find ecstasy in living -<cr><lf>
the mere sense of living<cr><lf>
is joy ^."<cr><lf>
```

You can also use a D command to delete the <cr><lf> between two lines to join them together. Remember that the <cr> and <lf> are each counted as a character.

K (Kill) Command

The K command kills or deletes whole lines from the memory buffer and takes the forms:

nK, -nK

where n is the number of lines to be deleted. A positive number kills lines after the CP. A negative number kills lines before the CP. When no number is specified, ED kills the current line. If the character pointer is at the beginning of the second line (as shown below),

```
 Emily Dickinson said,<cr><lf>
^"I find ecstasy in living -<cr><lf>
 the mere sense of living<cr><lf>
 is joy enough."<cr><lf>
```

then the command -K deletes the previous line and the memory buffer changes:

```
^"I find ecstasy in living -<cr><lf>
 the mere sense of living<cr><lf>
 is joy enough."<cr><lf>
```

If the CP is in the middle of a line, a K command kills only the characters from the CP to the end of the line and concatenates the characters before the CP with the next line. A -K command deletes all the characters between the beginning of the previous line and the CP. A 0K command deletes the characters on the line up to the CP.

You can use the special $ character to delete all the text from the CP to the beginning or end of the buffer. Be careful when using $K because you cannot reclaim lines after they are removed from the memory buffer.

Inserting Characters into the Memory Buffer

I (Insert) Command

To insert characters into the memory buffer from the screen, use the I command. Its forms are:

```
I
Istring^Z
```

When you type the first command, ED enters insert mode. In this mode, all keystrokes are added directly to the memory buffer. ED enters characters in lines and does not start a new line until you press the ENTER key.

```
A>ED B:QUOTE.TEX

NEW FILE
   : *i
  1:  Emily Dickinson said,
  2:  "I find ecstasy in living -
  3:  the mere sense of living
  4:  is joy enough."
  5:  ^Z
   : *
```

Note: To exit from insert mode, you must press CTRL-Z or ESC. When the ED prompt, *, appears on the screen, ED is not in insert mode.

In command mode, you can use CP/M-8000 line-editing control characters to edit your input. The table below lists these control characters.

Table 6-3. CP/M-8000 Line-editing Controls

Command	Result
CTRL-C	Abort the editor and return to the CP/M-8000 system.
CTRL-E	Return carriage for long lines without transmitting command line to the buffer.
CTRL-H	Delete the last character typed on the current line.
CTRL-U	Delete the entire line currently being typed.

Table 6-3. (continued)

Command	Result
CTRL-X	Delete the entire line currently being typed. Same as CTRL-U.
Rubout	Remove the last character and echo deleted character at the screen.

Note: In insert mode, the same line-editing controls exist except for CTRL-C and CTRL-E.

When entering a combination of numbers and letters, you might find it inconvenient to press the CAPS-LOCK key if your terminal translates caps-locked numbers to special characters. ED provides two ways to translate your alphabetic input to uppercase without affecting numbers. The first is to enter the insert command letter in uppercase: I. All alphabetics entered during the course of the capitalized command, either in insert mode or as a string, are translated to uppercase. (If you enter the insert command letter in lowercase, all alphabetics are inserted as typed). The second method is to enter a U command before inserting text. Uppercase translation remains in effect until you enter a -U command.

Istring^Z (Insert String) Command

The second form of the I command does not enter insert mode. It inserts the character string into the memory buffer and returns immediately to the ED prompt. You can use CP/M-8000's line editing control characters to edit the command string.

To insert a string, first use one of the commands that position the CP. You must move the CP to the place where you want to insert a string. For example, if you want to insert a string at the beginning of the first line, use a B command to move the CP to the beginning of the buffer. With the CP positioned correctly, enter an insert string, as shown below:

```
iIn 1870, ^Z
```

This inserts the phrase "In 1870, " at the beginning of the first line, and returns immediately to the ED prompt. In the memory buffer, the CP appears after the inserted string, as shown below:

```
In 1870, ^Emily Dickinson said,<cr><lf>
```

Replacing Characters

S (Substitute) Command

The S command searches the memory buffer for the specified string, and when the string is found, the S command automatically substitutes a new string for the search string. The S command takes the form:

nSsearch string^Znew string

where n is the number of substitutions to make. If no number is specified, ED searches for the next occurrence of the search string in the memory buffer. For example, the command:

Emily Dickinson^ZThe poet

searches for the first occurrence of "Emily Dickinson" and substitutes "The poet." In the memory buffer, the CP appears after the substituted phrase, as shown below:

The poet^ said,<cr><lf>

If uppercase translation is enabled by a capital S command letter, ED looks for a capitalized search string and inserts a capitalized insert string. Note that if you combine this command with other commands, you must terminate the new string with a CTRL-Z.

Combining ED Commands

It saves keystrokes and editing time to combine the editing and display commands. You can type any number of ED commands on the same line. ED executes the command string only after you press the carriage-return key. Use CP/M-8000's line-editing controls to manipulate ED command strings.

When you combine several commands on a line, ED executes them in the order they are entered, from left to right on the command line. There are four restrictions to combining ED commands:

- The combined-command line must not exceed CP/M-8000's 128-character maximum.
- If the combined-command line contains a character string, the line must not exceed 100 characters.
- Commands to terminate an editing session must not appear in a combined-command line.

- Commands, such as the I, J, R, S, and X commands, that require character strings or filespecs must be either the last command on a line or must be terminated with a CTRL-Z or ESC character, even if no character string or filespec is given.

While the examples in the previous section show the memory buffer and the position of the character pointer, the examples in this section show how the screen looks during an editing session. Remember that the character pointer is imaginary, but you must picture its location because ED's commands display and edit text in relation to the character pointer.

Moving the Character Pointer

To move the CP to the end of a line without calculating the number of characters, combine an L command with a C command, L-2C. This command string accounts for the <cr><lf> sequence at the end of the line.

Change the C command in this command string to move the CP more characters to the left. You can use this command string if you must make a change at the end of the line and you do not want to calculate the number of characters before the change, as in the following example.

```
    1: *T
    1:  Emily Dickinson said,
    1: *L-7CT
said,
    1: *
```

Displaying Text

A T command types from the CP to the end of the line. To see the entire line, you can combine an L command and a T command. Type 0LT to move the CP from the middle to the beginning of the line and then display the entire line. In the example below, the CP is in the middle of the line. 0L moves the CP to the beginning of the line. T types from the CP to the end of the line, allowing you to see the entire line.

```
    3: *T
sense of living
    3: **0LT
    3:  the mere sense of living
    3: *
```

The command 0TT displays the entire line without moving the CP.

To verify that an ED command moves the CP correctly, combine the command with the T command to display the line. The following example combines a C command and a T command.

```
2: *8CT
ecstasy in living -
2: *
```

```
4: *B#T
1:  Emily Dickinson said,
2:  "I find ecstasy in living -
3:  the mere sense of living
4:  is joy enough."
1: *
```

Editing

To edit text and verify corrections quickly, combine the edit commands with other ED commands that move the CP and display text. Command strings like the one below move the CP, delete specified characters, and verify changes quickly.

```
1: *15C5D0LT
1:  Emily Dickinson,
1: *
```

Combine the edit command K with other ED commands to delete entire lines and verify the correction quickly, as shown below.

```
1: *2L2KB#T
1:  Emily Dickinson said,
2:  "I find ecstasy in living -
1: *
```

The abbreviated form of the I (insert) command makes simple textual changes. To make and verify these changes, combine the I command string with the C command and the 0LT command string as shown below. Remember that the insert string must be terminated by a CTRL-Z.

```
1: *20Ci to a friend^Z0LT
1:  Emily Dickinson said to a friend,
1: *
```

Advanced ED Commands

The basic editing commands discussed above allow you to use ED for all your editing. The following ED commands, however, enhance ED's usefulness.

Moving the CP and Displaying Text

P (Page) Command

Although you can display any amount of text at the screen with a T command, it is sometimes more convenient to page through the buffer, viewing whole screens of data and moving the CP to the top of each new screen at the same time. To do this, use ED's P command. The P command takes the following forms:

nP, -nP

where n is the number of pages to be displayed. If you do not specify n, ED types the 23 lines following the CP and then moves the CP forward 23 lines. This leaves the CP pointing to the first character on the screen.

To display the current page without moving the CP, enter 0P. The special character 0 prevents the movement of the CP. If you specify a negative number for n, P pages backwards towards the top of the file.

n: (Line Number) Command

When line numbers are being displayed, ED accepts a line number as a command to specify a destination for the CP. The form for the line number command is

n:

where n is the number of the destination line. This command places the CP at the beginning of the specified line. For example, the command 4: moves the CP to the beginning of the fourth line.

Remember that ED dynamically renumbers text lines in the buffer each time a line is added or deleted. Therefore, the number of the destination line you have in mind can change during editing.

:n (Through Line Number) Command

The inverse of the line number command specifies that a command should be executed through a certain line number. You can use this command only with three ED commands: the T (type) command, the L (line) command, and the K (kill) command. The :n command takes the following form:

```
:ncommand
```

where n is the line number through which the command is to be executed. The :n part of the command does not move the CP, but the command that follows it might.

You can combine n: with :n to specify a range of lines through which a command should be executed. For example, the command 2::4T types the second, third, and fourth lines, as shown below.

```
1: *2::4T
2:  "I find ecstasy in living -
3:  the mere sense of living
4:  is joy enough."
2: *
```

Finding and Replacing Character Strings

ED supports a find command, F, that searches through the memory buffer and places the CP after the word or phrase you want. The N command allows ED to search through the entire source file instead of just the buffer. The J command searches for and then juxtaposes character strings.

F (Find) Command

The F command performs the simplest find function. Its form is

```
nFstring
```

where n is the occurrence of the string to be found. Any number you enter must be positive because ED can only search from the CP to the bottom of the buffer. If you enter no number, ED finds the next occurrence of the string in the file. In the following example, the second occurrence of the word "living" is found.

```
1: *2fliving
3: *
```

The character pointer moves to the beginning of the third line where the second occurrence of the word "living" is located. To display the line, combine the find command with a type

command. Note that if you follow an F command with another ED command on the same line, you must terminate the string with a CTRL-Z, as shown below.

```
1: *2fliving^Z01t
3:  the mere sense of living
```

It makes a difference whether you enter the F command in upper- or lowercase. If you enter F, ED internally translates the argument string to uppercase. If you specify f, ED looks for an exact match. For example, FCp/m-8000 searches for CP/M-8000, but fCp/m-8000 searches for Cp/m-8000 and cannot find CP/M-8000, or cp/m-8000.

If ED does not find a match for the string in the memory buffer, it issues the message:

```
BREAK "$" AT
```

where the symbol '$ indicates that the search failed during the execution of an F command.

N Command

The N command extends the search function beyond the memory buffer to include the source file. If the search is successful, it leaves the CP pointing to the first character after the search string. The form of the N command is

```
nNstring
```

where n is the occurrence of the string to be found. If no number is entered, ED looks for the next occurrence of the string in the file. Using an uppercase N in an N command has the same effect as an uppercase F in an F command. Note that if you follow an N command with another ED command, you must terminate the string with a CTRL-Z.

When an N command is executed, ED searches the memory buffer for the specified string. If ED does not find the string, it does not issue an error message. Instead, ED automatically writes the searched data from the buffer into the new file. Then ED performs a 0A command to fill the buffer with unsearched data from the source file. ED continues to search the buffer, write out data, and append new data until it either finds the string or reaches the end of the source file. If ED reaches the end of the source file, ED issues the following message:

```
BREAK  "$" AT
```

Because ED writes the searched data to the new file before looking for more data in the source file, ED usually writes the contents of the buffer to the new file before finding the end of the source file and issuing the error message.

Note: You must use the H command to continue an edit session after the source file is exhausted and the memory buffer is emptied.

J (Juxtapose) Command

The J command inserts a string after the search string, then deletes any characters between the end of the inserted string to the beginning of the third (delete-to) string. This juxtaposes the string between the search and delete-to strings with the insert string. The form of the J command is

nJsearch string^Zinsert string^Zdelete-to string

where n is the occurrence of the search string. If no number is specified, ED searches for the next occurrence of the search string in the memory buffer. In the following example, ED searches for the word "Dickinson" and inserts the phrase "told a friend" after it and then deletes everything up to the comma.

```
1: *#T
1:  Emily Dickinson said,
2:  "I find ecstasy in living -
3:  the mere sense of living
4:  is joy enough."
1: *jDickinson^Z told a friend^Z,
1: *0lt
1:  Emily Dickinson told a friend,
1: *
```

If you combine this command with other commands, you must terminate the delete-to string with a CTRL-Z or ESC. (This is shown in the following example). If an uppercase J command letter is specified, ED looks for uppercase search and delete-to strings and inserts an uppercase insert string.

The J command is especially useful when revising comments in assembly language source code, as shown below.

```
236:  SORT.   LXI    H, SW     ;ADDRESS TOGGLE SWITCH
236: *j;^ZADDRESS SWITCH TOGGLE^Z^L^ZOLT
236:  SORT    LXI    H, SW     ;ADDRESS SWITCH TOGGLE
236: *
```

In this example, ED searches for the first semicolon and inserts ADDRESS SWITCH TOGGLE after the mark and then deletes to the <cr><lf> sequence, represented by CTRL-L. (In any search string, you can use CTRL-L to represent a <cr><lf> when your desired phrase extends across a line break. You can also use a CTRL-I in a search string to represent a tab).

Note: If long strings make your command longer than your screen line length, enter a CTRL-E to cause a physical carriage return at the screen. A CTRL-E returns the cursor to the left edge of the screen, but does not send the command line to ED. Remember that no ED command line contains strings can exceed 100 characters. When you finish your command, press the carriage-return key to send the command to ED.

M (Macro) Command

An ED macro command, M, can increase the usefulness of a string of commands. The M command allows you to group ED commands together for repeated execution. The form of the M command is

```
nMcommand string
```

where n is the number of times the command string is to be executed. A negative number is not a valid argument for an M command. If no number is specified, the special character * is assumed, and ED executes the command string until it reaches the end of data in the buffer or the end of the source file, depending on the commands specified in the string. In the following example, ED executes the four commands repetitively until it reaches the end of the memory buffer:

```
1: "mfliving^Z-6diLiving^Z0lt
2:  "I find ecstasy in Living -
3:  the mere sense of Living

BREAK "$" AT ^Z
3: *
```

The terminator for an M command is a carriage return; therefore, an M command must be the last command on the line. Also, all character strings that appear in a macro must be terminated by CTRL-Z or ESC. If a character string ends the combined-command string, it must be terminated by CTRL-Z, then followed by a <cr> to end the M command.

The execution of a macro command always ends in a BREAK "#" message, even when you have limited the number of times the macro is to be performed, and ED does not reach the end of the buffer or source file. Usually the command letter displayed in the message is one of the commands from the string, not M.

To abort a macro command, press CTRL-C at the keyboard.

Moving Text Blocks

To move a group of lines from one area of your data to another, use an X command to write the text block into a temporary LIB file, then a K command to remove these lines from their original location, and finally an R command to read the block into its new location.

X (Transfer) Command

The X command takes the forms:

```
nX
nX filespec^Z
```

where n is the number of lines from the CP towards the bottom of the buffer that are to be transferred to a temporary file; therefore, n must always be a positive number. If no filename is specified, X$$$$$$$ is assumed. If no filetype is specified, LIB is assumed. If the X command is not the last command on the line, the command must be terminated by a CTRL-Z or ESC. In the following example, just one line is transferred to the temporary file:

```
1: *X
1: *t
1:  Emily Dickinson said,
1: *kt
1:  "I find ecstasy in living -
1: *
```

If no library file is specified, ED looks for a file named X$$$$$$$.LIB. If the file does not exist, ED creates it. If a previous X command already created the library file, ED appends the specified lines to the end of the existing file.

Use the special character 0 as the n argument in an X command to delete any file from within ED.

R (Read) Command

The X command transfers the next n lines from the current line to a library file. The R command can retrieve the transferred lines. The R command takes the forms:

```
R
Rfilespec
```

If no filename is specified, X$$$$$$$ is assumed. If no filetype is specified, LIB is assumed. R inserts the library file in front of the CP; therefore, after the file is added to the memory buffer, the CP points to the same character it did before the read, although the character is on a new line number. If you combine an R command with other commands, you must separate the filename from subsequent command letters with a CTRL-Z, as in the following example. Here ED types the entire file to verify the read.

```
1: *4l
: *R^ZB$T
1:  "I find ecstasy in living -
2:  the mere sense of living
3:  is joy enough."
4:  Emily Dickinson said,
1: *
```

Saving or Abandoning Changes: ED Exit

You can save or abandon editing changes with the following three commands.

H (Head of File) Command

An H command saves the contents of the memory buffer without ending the ED session, but it returns to the "head" of the file. It saves the current changes and lets you reedit the file without exiting ED. The form of the H command is

```
H
```

followed by a carriage return.

To execute an H command, ED first finalizes the new file, transferring all lines remaining in the buffer and the source file to the new file. Then ED closes the new file, erases any BAK file that has the same file specification as the original source file, and renames the original source file filename.BAK. ED then renames the new file, which has had the filetype $$$, with the original file specification. Finally, ED opens the newly renamed file as the new source file for a new edit, and opens a new $$$ file. When ED returns the * prompt, the CP is at the beginning of an empty memory buffer.

If you want to send the edited material to a file other than the original file, use the following command:

```
A>ED filespec differentfilespec
```

If you then restart the edit with the H command, ED renames the file differentfilename.$$$ to differentfilename.BAK and creates a new file of differentfilespec when you finish editing.

O (Original) Command

An O command abandons changes made since the beginning of the edit and allows you to return to the original source file and begin reediting without ending the ED session. The form of the O command is

```
O
```

followed by a carriage return. When you enter an O command, ED confirms that you want to abandon your changes by asking:

```
O (Y/N)?
```

You must respond with either a Y or an N; if you press any other key, ED repeats the question. When you enter Y, ED erases the temporary file and the contents of the memory buffer. When the * prompt returns, the character pointer is pointing to the beginning of an empty memory buffer, just as it is when you start ED.

Q (Quit) Command

A Q command abandons changes made since the beginning of the ED session and exits ED. The form of the Q command is

```
Q
```

followed by a carriage return.

When you enter a Q command, ED verifies that you want to abandon the changes by asking:

Q (Y/N)?

You must respond with either a Y or an N; if you press any other key, ED repeats the question. When you enter Y, ED erases the temporary file, closes the source file, and returns control to CP/M-8000.

Note: You can enter a CTRL-C to immediately return control to CP/M-8000. This does not give ED a chance to close the source or new files, but it prevents ED from deleting any temporary files.

ED Error Messages

ED returns one of two types of error messages: an ED error message if ED cannot execute an edit command, or a CP/M-8000 error message if ED cannot read or write to the specified file.

The form of an ED error message is

BREAK "x" AT c

where x is one of the symbols defined in the following table and c is the command letter where the error occurred.

Table 6-4. ED Error Symbols

Symbol	Meaning
#	Search failure. ED cannot find the string specified in an F, S, or N command.
?c	Unrecognized command letter c. ED does not recognize the indicated command letter, or an E, H, Q, or O command is not alone on its command line.
O	No LIB file. ED did not find the LIB file specified in an R command.
>	Buffer full. ED cannot put any more characters in the memory buffer, or string specified in an F, N, or S command is too long.
E	Command aborted. A keystroke at the keyboard aborted command execution.
F	File error. Followed by either DISK FULL or DIRECTORY FULL.

The following examples show how to recover from common editing error conditions. For example,

```
BREAK ">" AT A
```

means that ED filled the memory buffer before completing the execution of an A command. When this occurs, the character pointer is at the end of the buffer and no editing is possible. Use the OW command to write out half the buffer or use an O or H command and reedit the file.

```
BREAK "#" AT F
```

means that ED reached the end of the memory buffer without matching the string in an F command. At this point, the character pointer is at the end of the buffer. Move the CP with a B or n: line number command to resume editing.

If you get the following error message,

```
BREAK "F" AT F
DISK FULL
```

use the OX command to erase an unnecessary file on the disk or a B#Xd:buffer.sav command to write the contents of the memory buffer onto another disk.

If you see

```
BREAK "F" AT n
DIRECTORY FULL
```

use the same commands described in the previous message to recover from this file error.

The following table defines the disk file error messages ED returns when it cannot read or write a file.

Table 6-5. ED Disk File Error Messages

Message	Meaning
BDOS ERR ON d: RO	Disk d: has Read-Only attribute. This occurs if a different disk has been inserted in the drive since the last cold or warm boot.
** FILE IS READ ONLY **	The file specified in the command to invoke ED has the RO attribute. ED can read the file so that you can examine it, but ED cannot change a Read-Only file.

End of Section 6

Appendix A

Error Messages

Error messages come from several different sources. Each utility supplied with CP/M-8000 has its own set of error messages. These messages are displayed when there are errors in the utility command lines. CP/M-8000 also displays error messages when there are errors in calls to its built-in system modules: the Basic Disk Operating System (BDOS), the Basic Input/Output System (BIOS), and the Console Command Processor (CCP). The BDOS provides functions that access the file system, the BIOS provides functions that interface the peripheral device drivers for I/O processing, and the CCP provides the user interface that parses the user command line.

The utility error messages and the CP/M-8000 system error messages are listed in Table A-1 in alphabetic order with explanations and suggested user responses. The utility or system module that returns the error message is indicated in the explanation. If you are running an application program, you might see messages other than those listed here. Check the application program's documentation for explanations of those messages.

Table A-1. CP/M-8000 Error Messages

Message	Meaning
BAD DIRECTORY ON D: SPACE ALLOCATION CONFLICT	STAT error. This message is followed by a list of one or more filenames. The files listed have been doubly allocated and contain data blocks that are already allocated to another file on the disk. The error can be caused by either a hardware or software failure.

Table A-1. (continued)

Message	Meaning
	If you have maintained a backup disk with copies of your files, erase the affected files and reboot CP/M-8000. Rebooting the system cleans up the directory by reconstructing the directory allocation vectors. If you do not reboot the system the error recurs. To replace the files you erased, COPY or PIP the files from your backup disk. If you do not have a backup disk, you must recreate the doubly allocated files. If you can tell by careful checking which portions of the files are duplicated or deleted, you can recopy or PIP the files from the doubly allocated disk onto a new disk. Use ED to recreate the deleted portions and to delete the parts that were added to the wrong files. To clean up the doubly allocated disk so that it can be used again, erase it completely (ERA *.*) and reboot CP/M-8000. The problem that caused this error might or might not recur. If you were running an undebugged program at the time the error occurred, check the program carefully for an error that might have caused the space allocation error. If you were running a program that has been debugged and has caused no problem before, check your hardware. The error might have been caused by a damaged drive or loose connection. If the problem persists, contact your technical support personnel or the place you purchased your system for further assistance.
Bad relocation information bits	CCP error. The message indicates that the file specified in the command line is not a valid executable command file, or that the file has been corrupted.

Table A-1. (continued)

Message	Meaning
	Ensure that the file is a command file. Section 4 of this manual describes the format of a command file. If the file has been corrupted, reassemble or recompile the source file, and relink it before you reenter the command line.
BIOS ERROR — DISK X NOT SUPPORTED	BIOS error. The disk drive indicated by the variable X is not supported by the BIOS. The BDOS supports a maximum of 16 drives, lettered A through P. Check the documentation provided by the manufacturer for your system configuration to find out which of the BDOS drives your BIOS implements. Specify the correct drive code and reenter the command line.
BIOS ERROR — Invalid Disk Status	BIOS error. The disk controller returned unexpected or incomprehensible information to the BIOS. Try the operation again. If the error persists, check the hardware. If the error does not come from the hardware, it is caused by an error in the internal logic of the BIOS. Contact the place you purchased your system for assistance. You should provide the following information: • Indicate which version of the operating system you are using. • Describe your system's hardware configuration. • Provide sufficient information to reproduce the error. Indicate which program was running at the time the error occurred. If possible, you should also provide a disk with a copy of the program.

Table A-1. (continued)

Message	Meaning
BIOS ERROR -- DISK X NOT SUPPORTED	BIOS error. The disk drive indicated b the variable X is not supported by th BIOS. The BDOS supports a maximum of 1 drives, lettered A through P. Check th documentation provided by the manufacture for your system configuration to find ou which of the BDOS drives your BIC implements. Specify the correct driv code and reenter the command line.
BIOS ERROR -- Invalid Disk Status	BIOS error. The disk controller returne unexpected or incomprehensible informatic to the BIOS. Try the operation again. I the error persists, check the hardware If the error does not come from th hardware, it is caused by an error in th internal logic of the BIOS. Contact th place you purchased your system fo assistance. You should provide th following information: • Indicate which version of the operatin system you are using. • Describe your system's hardwar configuration. • Provide sufficient information t reproduce the error. Indicate whic program was running at the time th error occurred. If possible, yo should also provide a disk with a cop of the program.

Table A-1. (continued)

Message	Meaning
CP/M Disk change error on drive x	Disk error. The disk in the drive indicated by the variable x is not the same disk the system logged in previously. When the disk was replaced, you did not enter a CTRL-C to log in the current disk. Therefore, when you attempted to write to, erase, or rename a file on the current disk, the BDOS set the drive status to Read-Only and warm booted the system. The current disk in the drive was not overwritten. The drive status was returned to Read-Write when the system was warm booted. Each time a disk is changed, you must type a CTRL-C to log in the new disk.
CP/M Disk file error: filename is read-only. Do you want to: Change it to read/write (C), or Abort (A)?	Disk error. You attempted to write to, erase, or rename a file whose status is Read-Only. Specify one of the options enclosed in parentheses. If you specify the C option, the BDOS changes the status of the file to Read-Write and continues the operation. The Read-Only protection previously assigned to the file is lost. If you specify the A option or a CTRL-C, the program terminates and CP/M-8000 returns the system prompt.
CP/M Disk read error on drive x Do you want to: Abort (A), Retry (R), or Continue with bad data (C)?	Disk error. This message indicates a hardware error. Specify one of the options enclosed in parentheses. Each option is described below.

Table A-1. (continued)

Message	Meaning	
	A or CTRL-C	Terminates the operation and CP/M-8000 returns the system prompt.
	R	Retries the operation. If the retry fails, the system reprompts with the option message.
	C	Ignores the error and continues program execution. Be careful if you use this option. Program execution should not be continued for some types of programs. For example, if you are updating a data base and receive this error, but continue program execution, you can corrupt the index fields and the entire data base. For other programs, continuing program execution is recommended. For example, when you transfer a long text file and receive an error because one sector is bad, you can continue transferring the file. After the file is transferred, review the file, and add the data that was not transferred due to the bad sector.
CP/M Disk write error on drive x Do you want to: Abort (A), Retry (R), or Continue with bad data (C)?		
	Disk error. This message indicates a hardware error. Specify one of the options enclosed in parentheses. Each option is described below.	
	A or CTRL-C	Terminates the operation and CP/M-8000 returns the system prompt.
	R	Retries the operation. If the retry fails, the system reprompts with the option message.

Table A-1. (continued)

Message	Meaning	
	C	Ignores the error and continues program execution. Be careful if you use this option. Program execution should not be continued for some types of programs. For example, if you are updating a data base and receive this error message but continue program execution, you can corrupt the index fields and the entire data base. For other programs, continuing program execution is recommended. For example, when you transfer a long text file and receive an error message because one sector is bad, you can continue transferring the file. After the file is transferred, review the file, and add the data that was not transferred due to the bad sector.
CP/M Disk select error on drive x Do you want to: Abort (A), Retry (R)		Disk error. There is no disk in the drive or the disk is not inserted correctly. Ensure that the disk is securely inserted in the drive. If you enter the R option, the system retries the operation. If you enter the A option or CTRL-C the program terminates and CP/M-8000 returns the system prompt.
CP/M Disk select error on drive x		Disk error. The disk selected in the command line is outside the range A through P. CP/M-8000 can support up to 16 drives, lettered A through P. Check the documentation provided by the manufacturer to find out which drives your particular system configuration supports. Specify the correct drive code and reenter the command line.

Table A-1. (continued)

Message	Meaning
DIRECTORY FULL	ED error. There is not enough directory space for the file being written. You can use the OX filespec command to erase any unnecessary files on the disk without leaving the editor. Alternatively, you can save the contents of the memory buffer on another disk with the command B$Xfilespec, where filespec is a file on a different drive. You can then quit the edit. If you reedit the file, the output should be placed on a different drive with the command ED filespec d: where d: is a valid drive name other than the drive containing the source file. You can read the file saved with the Rfilespec command. Note that part of the file might not be in the memory buffer when you save it (if you have not appended the whole file or if you have issued any W commands).
DISK FULL	ED or PIP error. In ED, the message indicates that there is not enough disk space for the output file. This error can occur on the W, E, H, or X commands. If it occurs with the X command you can repeat the command, prefixing the filename with a different drive. Otherwise, you can try the recovery methods described above under the DIRECTORY FULL error. In PIP, this message indicates that the destination disk is full.

Table A-1. (continued)

Message	Meaning
Disk select error on Destination Disk select error on Source ERROR	Disk or COPY error. The disk selected in the command line is not supported by CP/M-8000. CP/M-8000 can support up to 16 drives, lettered A through P. Check the documentation provided by the manufacturer to find out which drives your particular system configuration supports. Specify the correct drive code and reenter the command line.
ERROR: BAD PARAMETER	PIP error. The command contains an illegal parameter. Refer to Section 5 in this manual on the PIP command for an explanation and listing of the PIP command parameters. Reenter the command with a valid parameter.
ERROR: CLOSE FILE - [filespec]	PIP error. The output file specified in the previous PIP command cannot be closed. This message indicates a fatal error in the internal logic of PIP. Contact the place you purchased your system for assistance. You should provide the following information: • Indicate which version of the operating system you are using. • Describe your system's hardware configuration. • Provide sufficient information to reproduce the error. Indicate which program was running at the time the error occurred. If possible, you should also provide a disk with a copy of the program.

Table A-1. (continued)

Message	Meaning
ERROR: 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 existing file is deleted and replaced with the file specified in the PIP command. If you do not want to delete the existing file, type N and reenter the PIP command with a different destination file specification: either specify a new filename, or direct the destination file to a different disk. This message can be avoided by using the W option. Refer to Section 5 of this manual on the PIP command for a discussion and list of the PIP options.
ERROR: DISK FULL - {filespec}	PIP error. PIP cannot read the input file indicated by the variable, filespec. This message indicates a bad disk. Ensure that the disk is in place and try the operation again. If the error persists, the disk is bad and must be replaced. COPY or PIP as much of the bad disk as possible, or use your backup disk, if you maintained one, to replace the lost files.
ERROR: DISK WRITE - {filespec}	PIP error. The disk to which PIP is writing is full. The file indicated by the variable, filespec, was not copied. Reenter the PIP command line and redirect the output to another disk, if possible. If you do not have another disk available, you must erase unnecessary files, if any, or replace the disk before you reenter the PIP command line.

Table A-1. (continued)

Message	Meaning
ERROR: FILE NOT FOUND - {filespec}	PIP error. The input file indicated by the variable, filespec, does not exist, or the drive code or user number is incorrect. Ensure that you are typing the correct filename and drive code, and that you are in the correct user number or have specified the correct user number in the option brackets before you reenter the PIP command line.
ERROR: HEX RECORD CHECKSUM - {filespec}	PIP error. A hex record checksum error was encountered during the transfer of a hex file. This message indicates a corrupted or truncated hex file. Recopy or PIP the hex file from your backup disks, if possible. If you do not have a backup, you must erase the bad file and recreate it.
ERROR: INVALID DESTINATION	PIP error. The destination specified in your PIP command is illegal. You have specified an input device as a destination, or used a wildcard file specification. Respecify the PIP command with an output device or a specific file as the destination. Refer to Section 5 of this manual on the PIP command for a discussion of the valid PIP destinations.
ERROR: INVALID FORMAT	PIP error. The format of your PIP command is illegal. Refer to Section 5 of this manual on the PIP command for the correct syntax.

Table A-1. (continued)

Message	Meaning
ERROR: INVALID HEX DIGIT - {filespec}	PIP error. An invalid hex digit has been encountered while reading a hex file. This message indicates a truncated or corrupted hex file. Recopy or PIP the file from your backup disk. If you do not have a backup, you must erase the bad file and recreate it.
ERROR: INVALID SEPARATOR	PIP error. You have placed an invalid character for a separator between two input filenames. Reenter the PIP command line using a comma as the separator between the input filenames.
ERROR: INVALID SOURCE	PIP error. The source specified in your PIP command is illegal. You have specified an output device as a source or used a wildcard filename. Reenter the PIP command line with an input device or a specific filename as the source. Refer to Section 5 of this manual on PIP for a discussion of the legal sources for a PIP command.
ERROR: INVALID USER NUMBER	PIP error. You have specified a user number greater than 15. User numbers range from 0 to 15. Reenter the PIP command line using a valid user number.
ERROR: NO DIRECTORY SPACE - {filespec}	PIP error. The disk to which PIP is writing has no directory space available. Erase unnecessary files, if any, or replace the disk before you reenter the PIP command line.

Table A-1. (continued)

Message	Meaning
ERROR: OPEN FILE File Not Found - {filespec} ERROR: OPEN FILE [SYS]TEM FILE	PIP error. The first message indicates that PIP could not find the source file specified in the PIP command line. Check the directory of the appropriate drive and user number for the correct spelling of the file. The second message indicates that the source file specified in the PIP command line is a system file, that is, a file with the SYS attribute set. You must use PIP's R option to enable PIP to find a system file.
ERROR: QUIT NOT FOUND	PIP error. The string argument to a Q parameter was not found in your input file. Ensure that you typed the correct string and try the operation again. This error can also occur when entering the Q option from the CP/M-8000 command line. CP/M-8000 translates everything in the command line into uppercase. To avoid this problem, enter the Q option from inside PIP. Enter PIP, and wait for the * prompt before entering the file specifications and Q option.
ERROR: START NOT FOUND	PIP error. The string argument to an S parameter could not be found in the source file. Ensure that you are typing the correct string and try the operation again. This error can also occur when entering the S option from the CP/M-8000 command line. CP/M-8000 translates everything in the command line into uppercase. To avoid this problem, enter the S option from inside PIP. Enter PIP, and wait for the * prompt before entering the file specifications and S option.

Table A-1. (continued)

Message	Meaning
ERROR: UNEXPECTED END OF HEX FILE - {filespec}	PIP error. The last hex record was not as long as the character count indicated. The hex file is truncated. Recopy or PIP the truncated file from your backup disk. If you do not have a backup, you must erase the bad file and recreate it.
ERROR: USER ABORTED	PIP error. You aborted a PIP operation by entering a CTRL-C or by pressing a key. If you do not intend to abort the operation, you must reenter the PIP command line.
ERROR: VERIFY - {filespec}	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. This indicates a bad disk or a hardware error. Try the operation again. If the error persists, insert a new disk and reenter the PIP command line. If the error still persists, check the hardware.
Extraneous argument ignored: argument?	COPY error. The command line contains too many arguments. The extraneous arguments are indicated by the variable, arguments. The command was carried out up to, but not including, the extraneous arguments. Refer to Section 5 in this manual on COPY for the correct syntax of the COPY command line. Reenter the command line with the correct syntax for the remaining arguments.

Table A-1. (continued)

Message	Meaning
File already exists	REN error. The name specified in the command line as the new filename already exists. Use the ERA command to delete the existing file if you want to replace it with the new file. If not, select another filename and reenter the REN command line.
FILE IS READ/ONLY	ED error. A Read-Only file cannot be edited with the ED command, ED filespec. The command, ED inputfilespec outputfilespec, should be used instead.
FILE NOT FOUND **FILE NOT FOUND - {filespec}**	ED, STAT error. ED or STAT could not find the file indicated by the variable, filespec. Check the filename, drive code, and user number before you reenter the command line.
FILENAME REQUIRED	ED error. The ED command was typed without a filename. Reenter the ED command followed by the name of the file you want to edit or create.
Formatting Error	FORMAT error. This message indicates either a bad disk or a hardware error. Try the operation again. If the error recurs, replace the disk with a new disk. If the error persists, look for a hardware error. Contact your technical support personnel or the place you purchased your system for assistance.

Table A-1. (continued)

Message	Meaning
Insufficient memory or bad file header	CCP error. This error could result from one of three causes: • The file is not a valid executable command file. • The program is too large for the available memory. • The program is linked to an absolute location in memory that cannot be used. Ensure that you are requesting the correct file. This error can occur when you enter the filename before you enter the command for a utility. Check the appropriate section of this User's Guide or the CP/M-8000 Programmer's Guide for the correct command syntax before you reenter the command line. If you are trying to run a program when this error occurs, the program file might have been corrupted. Reassemble or recompile the source file and relink it before you reenter the command line. Add more memory boards to the system configuration, or rewrite the program to use less memory. The program must be made relocatable or linked to a usable memory location. The BDOS Get/Set TPA Limits Function (63) returns the high and low boundaries of the memory space available for loading programs. See the CP/M-8000 Programmer's Guide for more information on Function 63.

Table A-1. (continued)

Message	Meaning
INVALID ASSIGNMENT USE: STAT D:FILENAME.TYP [SIZE] [RO] [RW] [SYS] OR [DIR]	STAT error. An invalid file assignment was attempted. The error message is followed by the correct syntax and a list of the valid file assignments. Refer to Section 5 of this manual on the STAT command for a discussion of file status assignments. Specify a valid file assignment and reenter the STAT command line.
INVALID FILENAME	ED error. The filename specified is too long, or a delimiter was used as the first character. Filenames cannot be longer than eight characters. Refer to Section 2 of this manual on naming files for a list and description of the delimiters that cannot be used as the first character in a filename. Correct the filename and reenter the ED command line.
Invalid or missing drive code	FORMAT error. The FORMAT command line is either missing a drive code specification, or the drive specified is not supported by the BDOS. CP/M-8000 can support up to 16 drives, lettered A through P. Check the documentation provided by the manufacturer to find out which drives your particular system configuration supports. Specify the correct drive code and reenter the command line.
No file	CCP, COPY, ERA, REN error. The filename specified in the command line does not exist. Ensure that you use the correct filename and reenter the command line.

Table A-1. (continued)

Message	Meaning
NO MEMORY	ED error. There is not enough space in available memory to create the buffer that ED requires to edit a file. Contact your technical support personnel or place of purchase for assistance. This problem can occur when ED is absolutely linked too high in the TPA to allow for the buffer. ED needs 4K of memory at a location immediately higher in memory than itself for a buffer. Relocate ED to a lower location. If ED is not linked too high, reconfigure the TPA of your system, if possible, by eliminating some system extensions. If it is not possible to reconfigure the TPA, additional memory boards must be added to the system configuration to run ED.
No wildcard filenames	CCP, REN error. The command specified in the command line does not accept wildcards in file specifications. Retype the command line using a specific filename.
OUTPUT FILE EXISTS, ERASE IT	ED error. This error occurs when placing the destination file on a different disk than the source. The destination filename already exists on the receiving disk. The existing file should be erased or another disk selected to receive the output file.
Program Load Error	CCP error. This message indicates an undefined failure of the BDOS. Reboot the system and try again. If the error persists, then it is caused by an error in the internal logic of the BDOS. Contact the place you purchased your system for assistance. You should provide the following information:

Table A-1. (continued)

Message	Meaning
	• Indicate which version of the operating system you are using. • Describe your system's hardware configuration. • Provide sufficient information to reproduce the error. Indicate which program was running at the time the error occurred. If possible, you should also provide a disk with a copy of the program.
READ error -----> Track: t Sector: s	COPY error. This message indicates a bad disk. The variable, t, is replaced with the number of the track where the error occurred. The variable, s, is replaced with the number of the bad sector. Try the operation again. If the error persists, use a new disk and COPY or PIP as many files as possible from the damaged disk. If you cannot COPY or PIP a file, you must replace the file from your backup disk. If you have not maintained a backup disk, you must recreate the lost file. Discard the damaged disk.
read error on program load	CCP error. This message indicates a premature end-of-file. The file is smaller than the header information indicates. Either the file header has been corrupted or the file was only partially written. If the program is one you have written, you must reassemble or recompile the source file and relink it before you reenter the command line. If this message occurs at any other time, and you have a backup copy of the program you were running, erase the program file and replace it from your backup files. If the error recurs, contact your technical support personnel or the place you purchased your system for assistance in locating the cause of the error.

Table A-1. (continued)

Message	Meaning
READ ONLY DRIVE STATUS NOT ALLOWED	STAT error. Read-Only drive status is not supported by CP/M-8000. Assign each file on the disk Read-Only status individually.
Select Error	FORMAT error. This message indicates one of three errors: • The drive specified is not supported by your system. • The disk on the drive selected is not in place. • A hardware error. CP/M-8000 supports a maximum of 16 drives, lettered A through P. Check the documentation provided by the manufacturer to find out which of these drives your particular system configuration supports. Reenter the command line with a valid drive code specification. Check the disk. Ensure that it is securely in place and that the drive door is closed. If the disk is in place and you have selected a valid drive code, look for a hardware error. The error might have been caused by a loose connection or damaged drive. Contact your technical support personnel or the place you purchased your system for further assistance.
Source and Destination must be different	COPY error. The source and destination drives specified are the same. Reenter the COPY command line with a different drive for the source than for the destination.

Table A-1. (continued)

Message	Meaning
SUB file not found	**SUBMIT error. The file requested either does not exist or does not have a filetype of SUB. Ensure that you are requesting the correct file. Refer to Section 5 in this manual on SUBMIT for information on creating and using submit files.**
Syntax: REN newfile=oldfile	**REN error. The syntax of the REN command line is incorrect. The correct syntax is given in the error message. Enter the REN command followed by a space, then the new filename, followed immediately by an equals sign and the name of the file you want to rename.**
Too many arguments: argument?	**DIR, ERA, TYPE, USER error. The command line contains too many arguments. The extraneous arguments are indicated by the variable, argument. Refer to Section 5 in this manual for the correct syntax for the command. Specify only as many arguments as the command syntax allows and reenter the command line. Use a second command line for the remaining arguments, if appropriate. The command might be repeated on one command line when separated by an exclamation point; for example,** **DIR a:!DIR b:**

Table A-1. (continued)

Message	Meaning
TOO MANY FILES TOO MANY ENTRIES IN INDEX TABLE	STAT error. There is not enough memory for STAT to sort the files specified or more than 512 files were specified. Move some files to another user number. Otherwise, break the group of files into smaller groups before you reenter the STAT command line. For example, use a wildcard (*.typ) to run STAT on only those files with a certain filetype.
Undefined option	COPY error. The option specified in the COPY command line is unsupported. The only COPY options supported by CP/M-8000 are a, for automatic, and v, for verify. Correct and reenter the COPY command line.
User # range is [0-15]	CCP error. The user number specified in the command line is not supported by the BIOS. The valid range is enclosed in the square brackets in the error message. Specify a user number between 0 and 15 (decimal) when you reenter the command line.
Write Error	FORMAT error. This message indicates either a bad disk or a hardware error. Try the operation again. If the error recurs, replace the disk with a new disk. If the error persists, look for a hardware error. The error might have been caused by a loose connection or damaged drive. Contact your technical support personnel or the place you purchased your system for further assistance.

Table A-1. (continued)

ge	Meaning
er ----> Track: t Sector: s	COPY error. This message indicates a bad disk. The variable, t, is replaced with the number of the track where the error occurred. The variable, s, is replaced with the number of the bad sector. Try the operation again. If the error persists, use a new disk and copy as many files as possible from the damaged disk. If you cannot copy a file, you must replace the file from your backup disk. If you have not maintained a backup disk, you must recreate the lost file. Discard the damaged disk.

End of Appendix A

Appendix B

Filetypes

CP/M-8000 identifies every file by a unique file specification, which consists of a drive specification, a filename, a filetype, and an optional password. The filetype is an optional three-character ending separated from the filename by a period. The filetype generally indicates a special kind of file. The following table lists common filetypes and their meanings.

Table B-1. Common Filetypes

Type	Meaning
8KN	Assembly language source file; the CP/M-8000 Assembler assembles or translates an ASM file into machine language.
8KS	Segmented assembly language source file.
BAK	Backup file created by text editor; the editor renames the source file with this filetype to indicate that the original file has been processed. The original file stays on disk as the backup file, so you can refer to it.
C	Program source file written in the C language.
Z8K	Z8000 executable program file.
H	Header files for C language programs.
HEX	Program file in hexadecimal format.
O	Compiled or assembled object files.
PRN	Printable file, displayable on console or printer.
TMP	AR8K temporary file used while archiving new libraries. AR8K, the CP/M-8000 Assembler, is explained in the CP/M-8000 Programmer's Guide.

Table B-1. (continued)

Type	Meaning
SUB	Filetype required for submit file containing one or more CP/M-8000 commands. The SUBMIT program executes commands in files of type SUB, providing a batch execution mode for CP/M-8000.
SYS	System file for CP/M-8000.
$$$	Temporary file, inaccessible to user.

End of Appendix B

Appendix C

CP/M-8000 Control Character Summary

Table C-1. CP/M-8000 Control Characters

Keystroke	Action
CTRL-C	Terminates executing transient utility program. Also terminates DIR, DIRS, SUBMIT, and TYPE built-in programs in progress. When entered as first character in a command line or during the execution of the above-mentioned programs, CTRL-C causes a warm start and the redisplay of the system prompt. (See Section 4 for a discussion of warm start..)
CTRL-E	Forces a physical carriage return but does not send the command line to CP/M-8000. Moves the cursor to the beginning of the following line without erasing your previous input.
CTRL-H	Deletes a character and moves the cursor left one character position.
CTRL-I	Moves the cursor to the next tab stop. Tab stops are automatically set at each eighth column. Has the same effect as the TAB key.
CTRL-J	Moves the cursor to the left of the current line and sends the command line to CP/M-8000. Has the same effect as a carriage return keystroke.
CTRL-M	Moves the cursor to the left of the current line and sends the command line to CP/M-8000. Has the same effect as a carriage return keystroke.

Table C-1. (continued)

Keystroke	Action
CTRL-P	Echoes all console activity to the printer. You can use CTRL-P after you halt scrolling with CTRL-S. A second CTRL-P ends printer echo. If your computer is not connected to a printer, typing a CTRL-P can cause your system to hang, requiring a cold start, that is, a complete resetting of your hardware and software. (See Section 1 for a description of a cold start.) CTRL-Q Restarts screen scrolling after a CTRL-S.
CTRL-R	Displays "#" at the current cursor location, moves the cursor to the next line, and retypes any partial command you have typed so far.
CTRL-S	Stops screen scrolling. If a display scrolls by too fast for you to read it, type CTRL-S. CTRL-Q restarts screen scrolling.
CTRL-U	Discards all the characters in your command line, displays "#" at the current cursor position, and moves the cursor to the next command line.
CTRL-X	Discards all the characters in the command line that you typed so far and moves the cursor back to the beginning of the current line.

End of Appendix C

Appendix D

FORMAT Command

Syntax: FORMAT {disk drive:}

Explanation: Digital Research provides a format utility, called FORMAT, that is compatible with CP/M-8000 and the Olivetti® M-20 computer. The Olivetti M-20 hardware is described in the CP/M-8000 Operating System System Guide. This utility formats 5-1/4-inch floppy diskettes only.

Formatting establishes the desired disk density (double density), the desired sector size (256 bytes per sector), and the number of tracks (40) appropriate for the disk. Formatting makes a disk ready to receive new data from another disk having the identical format. Thus, formatting is a preliminary, and often necessary, step to copying.

Although you can often set the density and sector size by using a formatting utility, you have no control over the number of tracks that are selected by the utility. That parameter is determined by the particular disk controller logic inside your disk cabinet; it is hardware-dependent. Consequently, computer hardware manufacturers often write the formatting utilities to go with the hardware they sell.

Whether you format a floppy diskette or a hard disk, all files on the disk are erased as part of the formatting procedure. The utility's behavior depends on the version supplied you by the manufacturer of your equipment.

Examples: To execute FORMAT at the CP/M-8000 command level, use the form:

A>**FORMAT** d:

where d: specifies a drive in the range a...p. Take care to include the colon following the drive letter.

After you press RETURN, FORMAT responds with the following prompt:

```
**********************************************************
*                                                        *
*                                                        *
*     Warning:  This will destroy all files on disk d:!  *
*                                                        *
*                                                        *
**********************************************************
```

Do you wish to continue?

Press Y for yes, in either upper- or lowercase, if you want to continue. Reply with N for no, if you decide not to format disk d. If you reply to the above question with any character other than Y, FORMAT aborts execution and returns you to the command level of CP/M-8000.

If you respond with a Y, FORMAT prompts:

```
Insert disk to be formatted in drive d: and press RETURN.
```

After inserting a diskette in the appropriate drive and pressing RETURN, FORMAT begins formatting the diskette and displays the message, "Formatting track n", where n is the track number currently being formatted.

End of Appendix D

Glossary

ambiguous filename: Filename that contains either of the CP/M-8000 wildcard characters, ? or *, in the primary filename or the filetype or both. When you replace characters in a filename with these wildcard characters, you create an ambiguous filename, and can easily reference more than one CP/M-8000 file in a single command line.

applications program: Program that needs an operating system to provide an environment in which to execute. Typical applications programs are business accounting packages, word processing (editing) programs, and mailing list programs.

argument: Symbol, usually a letter, indicating a place into which you can substitute a number, letter, or name to give an appropriate meaning to the formula in question.

ASCII: The American Standard Code for Information Interchange is a standard code for the representation of numbers, letters, and symbols. An ASCII text file is a file that can be intelligibly displayed or printed on paper.

attribute: File characteristic that can be set to ON or OFF.

backup: Copy of a disk or file made for safe keeping; also used as a verb, to back up, meaning to create a copy of a disk or file.

BDOS: Basic Disk Operating System--a CP/M-8000 operating system module that directs all activities associated with the file system.

BIOS: Basic Input/Output System--a CP/M-8000 operating system module that provides the interface functions associated with input/output processes.

bit: Switch in memory that can be set to ON (1) or OFF (0). Bits are grouped into bytes.

block: Disk storage area reserved for a specific use.

bootstrap: Process of loading an operating system into memory. Bootstrap procedures vary from system to system. The boot for an operating system must be customized for the memory size and hardware environment that the operating system manages. Typically, the boot is loaded automatically and executed at power up or when the computer is reset. Sometimes called a cold start.

buffer: Area of memory that temporarily stores data during the transfer of information.

built-in commands: Commands that permanently reside in memory. They respond quickly because they are not accessed from a disk.

byte: Unit of memory or disk storage containing eight bits.

command: Elements of a CP/M-8000 command line. In general, a CP/M-8000 command has three parts: the command keyword, the command tail, and a carriage return.

command file: Series of coded machine-executable instructions stored on disk as a program file, invoked in CP/M-8000 by typing the command keyword next to the system prompt on the console. The CP/M-8000 command files generally have a filetype of Z8K. Files are either command files or data files. Same as a command program.

command keyword: Name that identifies a CP/M-8000 command, usually the primary filename of a file of type Z8K, or a built-in command. The command keyword precedes the command tail and the carriage return in the command line.

command syntax: Statement that defines the correct way to enter a command. The correct structure generally includes the command keyword, the command tail, and a carriage return. A syntax line usually contains symbols that you should replace with actual values when you enter the command.

command tail: Part of a command that follows the command keyword in the command line. The command tail can include a drive specification, a filename or filetype, and options or parameters. Some commands do not require a command tail.

concatenate: PIP operation that copies two or more separate files into one new file in a specified sequence.

console: Primary input/output device. The console consists of a listing device, such as a screen and a keyboard, through which you communicate with the operating system or applications program.

control character: Nonprinting character combination that sends a simple command to CP/M-8000. Some control characters perform line-editing functions. To enter a control character, hold down the CONTROL key and strike the character key specified.

cursor: One-character symbol that can appear anywhere on the console screen. The cursor indicates the position where the next keystroke at the console will have an effect.

data file: Nonexecutable collection of similar information that generally requires a command file to manipulate it.

default: Currently selected disk drive and user number. Any command that does not specify a disk drive or a user number references the default disk drive and user number. When CP/M-8000 is first invoked, the default disk drive is drive A, and the default user number is 0 (until changed with the USER command).

delimiter: Special characters that separate different items in a command line. For example, in CP/M-8000, a colon separates the drive specification from the filename. A period separates the filename from the filetype. Brackets separate any options from their command or file specification. Commas separate one item in an option list from another. All of these special characters are delimiters.

directory: Portion of a disk that contains entries for each file on the disk. In response to the DIR command, CP/M-8000 displays the filenames stored in the directory.

DIR attribute: File attribute. A file with the DIR attribute can be displayed by a DIR command.

disk, diskette: Magnetic media used to store information. Programs and data are recorded on a disk in the same way that music is recorded on a cassette tape. The term diskette refers to smaller capacity, removable floppy disks. Disk can refer to a diskette, a removable cartridge disk, or a fixed hard disk.

disk drive: Peripheral device that reads and writes on hard or floppy disks. CP/M-8000 assigns a letter to each drive under its control. For example, CP/M-8000 can refer to the drives in a four-drive system as A, B, C, and D.

editor: Utility program that creates and modifies text files. An editor can be used to create documents or source code files for computer programs. The CP/M-8000 editor is invoked by typing the command ED next to the system prompt on the console.

executable: Ready to be run by the computer. Executable code is a series of instructions that can be carried out by the computer. For example, the computer cannot execute names and addresses, but it can execute a program that prints all those names and addresses on mailing labels.

execute a program: To start a program. When a program is running, the computer is executing a sequence of instructions.

FCB: File Control Block.

file: A unique collection of characters, instructions, or data stored on a disk. You can create files on a disk.

File Control Block: Structure used for accessing files on disk. Contains the drive, filename, filetype, and other information describing a file to be accessed or created on the disk.

filename: Name assigned to a file. A filename can include a primary filename of 1-8 characters and a filetype of 0-3 characters. A period separates the primary filename from the filetype.

file specification: Unique file identifier. A complete CP/M-8000 file specification includes a disk drive specification followed by a colon (d:), a primary filename of 1 to 8 characters, a period, and a filetype of 0 to 3 characters. For example, b:example.tex is a complete CP/M-8000 file specification.

filetype: Extension to a filename. A filetype can be from 0 to 3 characters and must be separated from the primary filename by a period. A filetype can tell something about the file. Certain programs require that files to be processed have certain filetypes.

floppy disk: Flexible magnetic disk used to store information. Floppy disks come in 5-1/4 and 8 inch diameters.

hard disk: Rigid, platter-like magnetic disk sealed in a container. A hard disk stores more information than a floppy disk.

hardware: Physical components of a computer.

hex file: ASCII-printable representation of a command (machine language) file.

hexadecimal notation: Notation for the base 16 number system using the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F to represent the sixteen digits. Machine code is often converted to hexadecimal notation because it can be easily represented by ASCII characters, and therefore displayed or printed.

input: Data going into the computer, usually from an operator typing at the terminal or by a program reading from a disk.

interface: Object that allows two independent systems to communicate with each other, such as an interface between hardware and software in a microcomputer.

I/O: Abbreviation for input/output.

keyword: See command keyword.

kilobyte: 1024 bytes, denoted as 1K. 32 kilobytes equal 32K. 1024 kilobytes equal one megabyte, or over one million bytes.

list device: Device such as a printer onto which data can be listed or printed.

logged in: Made known to the operating system, in reference to drives. A drive is logged in when it is selected by the operator or an executing process. It remains selected or logged in until you change disks in a floppy disk drive or enter CTRL-C at the command level.

logical: Representation of something that might or might not be the same in its actual physical form. For example, a hard disk can occupy one physical drive, and yet you can divide its available storage so that it appears to the user as several different drives. These apparent drives are the logical drives.

megabyte: Over one million bytes; 1024 kilobytes. See byte, kilobyte.

microprocessor: Silicon chip that is the Central Processing Unit (CPU) of the microcomputer.

operating system: Collection of programs that supervises the execution of other programs and manages the computer's resources. An operating system provides an orderly input/output environment between the computer and its peripheral devices. It enables user-written programs to execute easily.

option: One of many parameters that can be part of a command tail. Use options to specifiy additional conditions for a command's execution.

output: Data that the processor sends to the console or disk.

parameter: Value in the command tail that provides additional information for the command. Technically, a parameter is a required element of a command.

peripheral devices: Devices external to the CPU. For example, terminals, printers, and disk drives are common peripheral devices that are not part of the processor, but are used in conjunction with it.

physical: Actual hardware of a computer. The physical environment varies from computer to computer.

primary filename: First 8 characters of a filename. The primary filename is a unique name that helps the user identify the file contents. A primary filename contains 1 to 8 alphanumeric characters and can include some special characters. The primary filename follows the optional drive specification and precedes the optional filetype.

program: Series of specially coded instructions that perform specific tasks when executed by a computer.

prompt: Any characters displayed on the video screen to help the user decide the next appropriate action. A system prompt is a special prompt displayed by the operating system. The system prompt indicates that the operating system is ready to accept your input. The CP/M-8000 system prompt is an alphabetic character followed by an angle bracket. The alphabetic character indicates the default drive. Some applications programs have their own special prompts.

Read-Only: Attribute that can be assigned to a disk file. Every file and drive is either Read-Only or Read-Write. When a file is set to Read-Only, you can read from that file but not write any changes to it. The default setting is Read-Write. Any error in resetting the disk or changing media automatically sets the entire drive to Read-Only until the error is corrected.

Read-Write: Attribute that can be assigned to a disk file. The Read-Write attribute allows you to read from and write to a specific file so marked. See **Read-Only**.

record: Collection of data. A file consists of one or more records stored on disk. An CP/M-8000 record is 128 bytes long.

RO: Abbreviation for **Read-Only**.

RW: Abbreviation for **Read-Write**.

sector: Portion of a disk track. There are a specified number of sectors on each track.

software: Specially coded programs that transmit machine-readable instructions to the computer, as opposed to hardware—the actual physical components of a computer.

source file: ASCII text file that is an input file for a processing program, such as an editor, text formatter, or assembler.

syntax: Format for entering a given command.

system attribute: A file attribute. You can give a file the system attribute by using the SYS option in the STAT command. A file with the SYS attribute is not displayed in response to a DIR command; you must use DIRS to view a directory of files having the system attribute.

system prompt: Symbol displayed by the operating system indicating that the system is ready to receive input. See **prompt**.

terminal: See **console**.

track: Concentric rings dividing a disk. There are 77 tracks on a typical eight-inch floppy disk.

turn-key application: Application designed for th computer-oriented user. For example, a typical tu application is designed so that the operator needs only t on the computer, insert the proper program disk and sele desired procedure from a selection of functions displayed on the screen.

upward-compatible: Term meaning that a program created f previously released operating system (or compiler, etc. under the newly released version of the same operating sy

user number: Number assigned to files in the disk directo that different users working from the same disk need deal with their own files and have their own directories. In 8000, files can be divided into 16 user groups (0-15).

utility: Tool. Program that enables the user to pe certain operations, such as copying files, erasing files editing files. Utilities are created for the convenien programmers and users.

wildcard characters: Special characters that match ce specified items. In CP/M-8000 there are two wil characters, ? and *. The ? can be substituted for any s character in a filename, and the * can be substituted fo primary filename or the filetype or both. By placing wil characters in filenames, you can create an ambiguous fil and quickly reference one or more files.

End of Glossary

Index

$$$ file, 5-16
:n (Through Line Number) Command, 6-18

A

A (Append) Command, 6-4
abort a copy operation, 5-20
access modes, 5-33, 5-34, 5-36
active
  status, 5-31
  user numbers, 5-37
assign physical device to logical device, 5-38
attributes, 5-33
AUI:, 3-3, 5-20
AUO, 3-3, 5-20

B

B (Beginning/Bottom) Command, 6-8
backup disks, 1-4
basic editing commands, 6-7
booting the system, 1-1
built-in commands, 4-1
bytes, 5-34

C

C (Character) Command, 6-8, 6-15
character
  pointer, 6-6
  transfer, 5-19
cold start, 1-1
combined-command line, 6-14
combining files, 5-19
command
  description, 4-10
  keyword, 1-2, 4-10
  line, 1-2
  mode, 6-12
  tail, 1-2
CON:, 3-3, 5-20
concatenation, 5-19
control character, 1-2
copy, 1-4, 1-5
COPY command, 5-2
copy files to and from auxiliary devices, 5-20
CP, 6-6
CTRL key, 1-2
CTRL-E, 6-21
CTRL-Z, 5-19, 5-21, 6-12, 6-20
cursor, 1-2

D

D (Delete) Command, 6-10
data files, 2-1
default
  drive, 2-3
  user number, 5-18
dest-filespec, 4-13
DEV:, 5-38
device names, 5-21
DIR, 5-6, 5-33, 5-36
  (Directory) Command, 1-2, 5-6
directory, 2-1, 2-6
  attribute, 5-33
  entries, 5-34
  space, 2-6
  verification, 5-34
DIRS (System Directory) Command, 5-6
disk
  free space, 5-32
  space allocation, 5-34
display
  scrolling, 3-1
  disk status, 5-37
Dn (delete characters) option, 5-24
drive, 2-7
  specifier, 2-2, 2-4, 4-7

E

E (echo) option, 5-17, 5-24
E (Exit) Command, 6-5
ED, 2-1
  Command, 5-8
  Command Summary, 5-9
  Error Symbols, 6-25
  messages, 6-2, 6-19, 6-25, 6-27
  prompt, 6-2, 6-12
  syntax, 6-1
editing, 6-16
editors, 6-1

end-of-file character, 5-19
EOF:, 5-20
ERA Command, 5-13
Examples, 4-10
executable file, 2-1

F

F (filter form-feeds)
option, 5-24
F (Find) Command, 6-18
FCB, 5-34
file, 2-1
attributes, 5-16
categories, 2-2
concatenation, 5-19
specification, 4-7, 5-34
filename, 4-7
filetype, 2-2, 4-7

G

Gn option, 5-15, 5-16,
5-19, 5-24,

H

H (Head of File) Command, 6-23.
H (hex data transfer)
option, 5-25

I

I (ignore) option, 5-25
I (Insert) Command, 6-12
input devices, 3-3
input-output port, 5-39
insert mode, 6-12
Istring^Z (Insert String)
Command, 6-13

J

J (Juxtapose) Command, 6-20

K

K (Kill) Command, 6-11
kilobytes, 5-34

L

L (Line) Command, 6-8, 6-15
L (translate case)
option, 5-25

line
editing controls, 6-13
numbers, 6-4
line-editing control
characters, 3-2
loading CP/M-8000, 1-1
logical
device, 3-3, 5-20, 5-38
long form of PIP, 5-17
LST:, 3-3, 5-20

M

M (Macro) Command, 6-21
memory buffer, 6-2, 6-3
multiple
command mode, 5-23
file copy, 5-18

N

N (add line numbers)
option, 5-25
n (Number) Command, 6-9
N Command, 6-19
n: (Line Number) Command, 6-17
NUL:, 5-20
numeric argument, 6-6

O

O
(object file transfer)
option, 5-25
(Original) Command, 6-24
option, 5-19
on-line
disk, 2-7
status, 5-37
Options, 4-10
output devices, 3-3

P

P (Page) Command, 6-17
physical device
drivers, 5-39
names, 5-38
PIP, 2-1,
command, 5-15
messages, 5-18
options, 5-16, 5-23

Pn (set page length)
option, 5-26
pound sign argument, 6-6
PRN:, 5-21
program file, 2-1

Q

Q (Quit) Command, 6-24
Qs (quit copying) option, 5-26

R

R (read system files)
option, 5-26
R (Read) Command, 6-23
Read-Only, 5-31
attribute, 5-33
files, 5-16
Read-Write, 5-31
attribute, 5-33
ready status, 2-7
real file size, 5-34
records, 5-33
recovering from editing error
conditions, 6-26
Recs, 5-33
REN
command, 5-29
messages, 5-30
reset, 2-7
RETURN key, 1-2
RO, 5-31, 5-33, 5-36
RW, 5-31, 5-33, 5-36

S

set file access modes, 5-36
short form of PIP, 5-17
single file copy, 5-15
SIZE, 5-34
source file, 6-2, 6-3
square brackets, 5-16, 5-23
Ss (start copying)
option, 5-26
STAT
command, 5-31
messages, 5-32, 5-34
syntax, 4-10
notation, 4-12
SYS, 5-33, 5-36
system
attribute, 5-33
prompt, 1-2, 2-7
reset, 1-1

T

T (Type) Command, 6-9,
6-15, 6-16
tab characters, 5-45
temporary
file, 5-16
work file, 6-2
text editor, 2-1
text transfer commands, 6-4
Tn (expand tabs) option, 5-27
transient commands, 4-1
TYPE
Command, 5-45
messages, 5-45

U

U (translate case) option, 5-26
U (Uppercase Translation)
Command, 6-13
USER command, 5-47
user numbers, 5-37, 5-47
utility commands, 4-1

V

V (verify) option, 5-17, 5-27
VAL:, 5-38
version number, 1-1
virtual file size, 5-34

W

W
(write over) option, 5-27
(Write) Command, 6-5
wildcard, 4-8, 5-33
characters, 2-5

X

X (Transfer) Command, 6-22
X$$$$$$$.LIB file, 6-23

Z

Z (zero the parity bit)
option, 5-27
Z8K, 2-3