

SIEMENS

Betriebssystem

CONCURRENT CP/M-86[®]

**Benutzeranleitung
(User's Guide)**



COPYRIGHT

Copyright © 1983 by Digital Research. 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, Post Office Box 579, Pacific Grove, California, 93950.

This manual is, however, tutorial in nature. Thus, the reader is granted permission to include the example programs, either in whole or in part, in his or her own programs.

DISCLAIMER

Digital Research 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 reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Digital Research to notify any person of such revision or changes.

TRADEMARKS

CP/M and CP/M-86 are registered trademarks of Digital Research. ASM-86, Concurrent CP/M-86 and DDT-86 are trademarks of Digital Research. Intel is a registered trademark of Intel Corporation. DEC is a registered trademark of Digital Equipment Corporation.

The *Concurrent CP/M-86 Operating System User's Guide* was printed in the United States of America.

First Edition: May 1983

Foreword

Concurrent CP/M-86™ is a multitasking operating system that lets you make the most of your computer by doing two or more jobs at the same time. For example, you can edit one or more documents, review one, and print another. Simply switch from one task to another with the touch of a key. Concurrent CP/M-86 also lets you use many powerful software tools such as word processors, spreadsheet calculators, data base managers, and bookkeeping systems.

How This Book Is Organized

The User's Guide introduces you to Concurrent CP/M-86 and tells you how to use it. It assumes you have read the manuals supplied by the manufacturer of your computer system and that you are familiar with the parts of your computer. You should know how to set up and turn on your computer, and how to handle, insert, and store disks. However, this manual does not assume you have had a great deal of experience with computers.

Section 1 provides you with an overview of operating systems in general and Concurrent CP/M-86 in particular. In Section 1 you will also find explanations for some of the words and phrases used in this manual.

Section 2 develops the command concepts you need to understand and use the commands defined in Section 3.

Section 3 alphabetically lists all the commands Concurrent CP/M-86 understands and executes, shows the syntax, and demonstrates examples for each command.

Section 4 describes ED, the Concurrent CP/M-86 text editor, in detail.

Appendix A lists the error messages Concurrent CP/M-86 displays when it encounters special conditions. Remedies are suggested for both specific error messages and general situations.

Appendix B is a glossary of computer terms that explains the terminology and concepts you encounter in learning about your computer.

Appendix C summarizes Concurrent CP/M-86 control characters.

Appendix D describes the various Concurrent CP/M-86 filetypes.

Appendix E summarizes the kinds of drive and file status information Concurrent CP/M-86 returns to you.

Appendix F is written primarily for those of you who have previous experience with the Digital Research family of operating systems; it describes the major differences between Concurrent CP/M-86 and other CP/M® operating systems.

If your Concurrent CP/M-86 is the first computer system you have used, you might find some of these topics confusing. Begin with Section 1 and read this manual section by section. It proceeds step-by-step, so you can easily understand how to use Concurrent CP/M-86.

Related Books

Concurrent CP/M-86 is supported and documented through four manuals:

- The *Concurrent CP/M-86 Operating System User's Guide*, which documents the user's interface to Concurrent CP/M-86, explains the various features used to execute applications programs and Digital Research utility programs.
- The *Concurrent CP/M-86 Operating System Programmer's Reference Guide*, which documents the applications programmer's interface to Concurrent CP/M-86, explains the internal file structure and system entry points—information essential for creating applications programs that run in the Concurrent CP/M-86 environment.
- The *Concurrent CP/M-86 Operating System Programmer's Utilities Guide* documents the Digital Research utility programs that programmers use to write, debug, and verify applications programs written for the Concurrent CP/M-86 environment.
- The *Concurrent CP/M-86 Operating System System Guide* documents the internal, hardware-dependent structures of Concurrent CP/M-86.

Note: *Concurrent CP/M-86 Operating System User's Guide* is hereafter cited as *Concurrent CP/M-86 User's Guide*, *Concurrent CP/M-86 Operating System Programmer's Reference Guide* is hereafter cited as *Concurrent CP/M-86 Programmer's Reference Guide*, *Concurrent CP/M-86 Operating System Programmer's Utilities Guide* is hereafter cited as *Concurrent CP/M-86 Programmer's Utilities Guide*, *Concurrent CP/M-86 Operating System System Guide* is hereafter cited as *Concurrent CP/M-86 System Guide*.

Table of Contents

1	Introduction to Concurrent CP/M-86	
1.1	What Is An Operating System?	1-1
1.2	What Is Concurrent CP/M-86?	1-2
1.2.1	Multitasking	1-2
1.2.2	Virtual Consoles	1-3
1.2.3	Switching Virtual Consoles	1-3
1.2.4	Status Line	1-4
1.3	Drives, Disks, Directories, and Files	1-4
1.3.1	Filespecs, File Attributes, and Filetypes	1-5
1.3.2	User Numbers	1-5
1.4	One Last Word	1-5
2	Let's Get Started	
2.1	Booting Concurrent CP/M-86	2-1
2.2	What's on the Disk?	2-2
2.3	What Next?	2-2
2.4	Drives And Devices	2-3
2.5	Types Of Files	2-4
2.5.1	Naming Files	2-4
2.5.2	Creating A File	2-6
2.5.3	Accessing Files	2-6
2.5.4	Start-up Files	2-7
2.6	Computer Punctuation	2-7
2.7	Wildcard Characters	2-8
2.8	Passwords	2-10
2.9	Organizing Your Files	2-10
2.9.1	User Numbers	2-10
2.9.2	File Attributes	2-11
2.10	How Are Files Stored On Disks?	2-12
2.11	Making Mistakes	2-12
2.12	Control Characters	2-13
2.13	Command Summary	2-15

Table of Contents (continued)

3 Concurrent CP/M-86 Command Summary	
3.1 Command Line Syntax and File Specifications	3-1
3.2 How Commands Are Described	3-4
The ABORT Command	3-7
The DATE Command	3-8
The DIR Command	3-10
The ED Commands	3-13
The ERA Command	3-14
The ERAQ Command	3-16
The HELP Command	3-17
Creating Customized HELP Files	3-18
The INITDIR Command	3-20
The PIP Command	3-22
Single File Copy	3-22
Multiple File Copy/Making Back-ups	3-25
Combining Files	3-26
Copy Files to and from Auxiliary Devices	3-27
Multiple Command Mode	3-29
Using Options With PIP	3-30
The PRINTER Command	3-35
The REN Command	3-36
The SDIR Command	3-38
SDIR Full Format	3-39
SDIR Size Format	3-40
SDIR Options	3-40
The SET Command	3-44
Assign a Label to the Drive	3-44
Enable/Disable Password Protection	3-44
Assign Password to the Label	3-45
Assign Passwords to Files	3-46
Set Password Protection Mode	3-46
Assign a Default Password	3-47
SET Time Stamp Options on Disk	3-48
Set FILE Attribute Bits	3-50
Additional SET Examples	3-50
The SHOW Command	3-51
The SUBMIT Command	3-54

Table of Contents (continued)

The TYPE Command	3-58
The USER Command	3-59
The VCMODE Command	3-61
4 ED, the Concurrent CP/M-86 Context Editor	
4.1 Introduction to ED	4-1
4.2 Starting ED	4-2
4.3 ED Operation	4-3
4.3.1 Appending Text into the Buffer	4-5
4.3.2 ED Exit	4-6
4.4 Basic Editing Commands	4-8
4.4.1 Moving the Character Pointer	4-10
4.4.2 Displaying Memory Buffer Contents	4-12
4.4.3 Deleting Characters	4-13
4.4.4 Inserting Characters into the Memory Buffer	4-15
4.4.5 Replacing Characters	4-17
4.5 Combining ED Commands	4-18
4.5.1 Moving the Character Pointer	4-18
4.5.2 Displaying Text	4-19
4.5.3 Editing	4-20
4.6 Advanced ED Commands	4-20
4.6.1 Moving the CP and Displaying Text	4-21
4.6.2 Finding and Replacing Character Strings	4-22
4.6.3 Moving Text Blocks	4-27
4.6.4 Saving or Abandoning Changes: ED Exit	4-29
4.7 ED Error Messages	4-31

Table of Contents (continued)

Appendixes

A	Concurrent CP/M-86 Error Handling	A-1
A.1	Concurrent CP/M-86 Error Messages	A-1
A.2	Common Error Handling Procedures	A-36
A.2.1	Checklist for Using Files	A-36
A.2.2	Corrupted Files	A-37
A.2.3	Hardware Errors Checklist	A-38
A.2.4	No Memory	A-39
A.2.5	GENCCPM Options	A-39
A.2.6	Reporting Logic Errors	A-40
B	Concurrent CP/M-86 User's Glossary	B-1
C	Concurrent CP/M-86 Control Character Summary	C-1
D	Filetypes	D-1
E	Drive and File Status Summary	E-1
F	Differences Between Concurrent CP/M-86 2.0 and CP/M-86	F-1

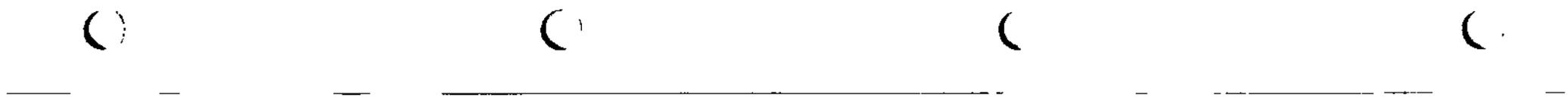
Table of Contents (continued)

Tables

2-1.	Special Characters	2-8
2-2.	Concurrent CP/M-86 Control Character Commands	2-13
2-3.	Command Summary	2-15
3-1.	File Specification Components	3-2
3-2.	Time and Date Specification Description	3-8
3-3.	PIP Options	3-31
3-4.	SDIR Options	3-40
3-5.	Password Protection Modes	3-47
4-1.	Text Transfer Commands	4-5
4-2.	Basic Editing Commands	4-9
4-3.	Concurrent CP/M-86 Line Editing Controls	4-16
4-4.	ED Error Symbols	4-31
4-5.	ED File Error Messages	4-33
A-1.	Concurrent CP/M-86 Error Messages	A-2
A-2.	Concurrent CP/M-86 Search Path	A-37
C-1.	Concurrent CP/M-86 Control Characters	C-1
D-1.	Filetypes	D-2
E-1.	Display File Size	E-1
E-2.	Display File Attributes	E-1
E-3.	Display Time Stamping and Protection Modes	E-2
E-4.	Ways to Display Diskette, Label, and System Status	E-2

Figures

3-1.	DIR Command and Command Tail	3-1
3-2.	File Specification Format	3-2
4-1.	Overall ED Operation	4-4
E-1.	Sample SHOW DRIVE Display	E-3
E-2.	Sample SHOW LABEL Display	E-3



Section 1

Introduction to Concurrent CP/M-86

This section gives you the information you need to use Concurrent CP/M-86. Written to help first-time users unravel the mysteries of computer jargon, this section briefly introduces you to operating systems and related concepts before it discusses Concurrent CP/M-86. In addition, this section also defines terminology used throughout this manual. If you are familiar with operating system concepts, the family of CP/M systems, other Concurrent CP/M-86 operating systems, and the difference between a bit and a byte, you might want to skip this section and begin with Section 2, "Let's Get Started."

1.1 What Is An Operating System?

Most of the appliances you deal with in daily living do one thing well. They are designed to fulfill a single purpose, such as toasting bread, drying hair, displaying a television picture, or calculating arithmetic functions. Because they perform a specific function, they do not require a great deal of skill to use—at most, turning a dial or pushing a few buttons.

A computer, on the other hand, does many things well. It can become a calculator, a word processor, an intelligent file cabinet, an opponent at chess, and many other things. You pay for this versatility by spending a certain amount of time telling the computer what to do; this is the art of computer programming. Once you have perfected a program for the computer, you can store it and refer to it later instead of repeating a long series of instructions. If you want to change the computer's operation, within its inherent limitations, you can revise your program as you wish. In a sense, as you gather a library of programs, the computer gets smarter and smarter.

You have purchased a computer program named Concurrent CP/M-86. It is a specific kind of computer program called an operating system. Operating systems store and retrieve data, such as sales figures, letters, or other information, and execute (run) other kinds of programs—programs known as applications programs, which turn the computer into a calculator, word processor, intelligent file cabinet, or opponent at chess. The operating system is a supervisor that manages your computer's resources at its most basic level.

Concurrent CP/M-86 performs the following three primary functions:

- Manages the input and output (I/O) from all devices—such as the console, keyboard, disk drives, and printer—attached to your computer. Input is the information your computer receives from an input device such as the keyboard. A system receives input when you type information from the keyboard or the operating system reads data from a disk. Output is the information the system sends to an output device such as a printer. Thus, your operating system provides the proper environment for your computer to respond to the information you enter at the keyboard or from a disk and to send the characters that you type to the console screen, the printer, or some other output device.
- Manages the files in which you store information on your computer system. By managing files, your system provides you with an orderly way to read (get information) or write (add information) to and from a disk. Your operating system also gives you information about files such as their name, type, and size.
- Allows you to load program files into your computer's memory and run them—whether they are utility programs, word processing, accounting, or other types of application programs that you purchase with your system or write yourself.

1.2 What Is Concurrent CP/M-86?

In addition to performing all the operating system tasks discussed previously, Concurrent CP/M-86 is an operating system with two unique features: multitasking and virtual consoles.

1.2.1 Multitasking

Multitasking gives you the extra resources you need to do more than one job concurrently. For example, you can work on one program while your computer works on others. While your system prints payroll checks and another unseen accounting program updates the payroll records, you can prepare a monthly progress report with a text processor.

1.2.2 Virtual Consoles

Concurrent CP/M-86 virtual consoles let you use one console screen as if it were more than one. Like a television set, which lets you switch between stations and view them one at a time on a single screen, with Concurrent CP/M-86, you can switch to and from several virtual consoles and view on a single console screen a program on a virtual console. These rules apply to using virtual consoles:

- You can only switch to and look at one virtual console at a time.
- The console screen displays output from a program on the current (the one you are looking at) virtual console only.

The following example illustrates these rules. On one console you are writing a letter, while other consoles perform tasks such as printing a payroll, or preparing a report. Unless you switch to another console, while you write the letter, all you see is the letter writing task. However, by pressing a key or a sequence of keys on your keyboard, you can switch to another console to check the progress of the payroll printing task or the report. Refer to the user information supplied by the manufacturer of your computer system for details on the keys you use to switch consoles.

1.2.3 Switching Virtual Consoles

The console you are working on is called the current or switched-in console, and the program is said to be operating in the foreground. When you switch away from a console, it is switched out, and its program is said to be operating in the background. The console screen is always assigned to the foreground program. Therefore, the console screen does not display the output of a background program. However, the output is not necessarily lost. A program in the background runs on a virtual console in one of two modes: dynamic or buffered.

When a console is in dynamic mode and the console is switched out, the operating system does not store the output of the program. Thus, dynamic mode is similar to switching stations on a television set; when you switch stations, programming continues but you cannot see it or the events that occur while the station is switched out.

Buffered mode lets you store what happens on a switched-out console. Concurrent CP/M-86 does not begin recording the program output until you switch your program into the background. For example, when the current console is in buffered mode, Concurrent displays output from the program on the console screen until you switch to another virtual console. While the program remains in the background, Concurrent CP/M-86 records, in a disk file, output from the program on the switched-out console. When you switch back to the console, Concurrent CP/M-86 displays on the console screen the output that was stored in the disk file. The output shows you what the program has done since you last switched the console out. Thus, buffered mode is similar to an instant replay on a television set, but you only see the scenes that happened while you were switched to another station.

To check or set the mode of a console, use the VCMODE command described in the Command Summary in Section 3.

1.2.4 Status Line

Your computer system might or might not have a status line. If your system has a status line, it tells you which console is current, its operating mode, and the number of the printer assigned to the console, as well as the name of the program currently running.

If any files in the system are open, the status line also tells you which drives have open files. An open file is a file in use by a program. If the open files are on floppy disks, you need this information so you do not remove the disk by mistake.

1.3 Drives, Disks, Directories, and Files

Drives, disks, directories, and files are analogous to a filing system in an office. A drive containing a disk is like a file cabinet full of individual file folders. One disk might contain all your outgoing correspondence, information for accounting reports, or the text of a document. Each disk has a directory that Concurrent CP/M-86 uses to locate files on disks in different drives. The directory is similar to the labels on the front of your file drawer. However, unlike the file drawer label, the directory of a disk contains the name of every file on the disk.

1.3.1 Filespecs, File Attributes, and Filetypes

Filespec, file attribute, and filetype are words you will see frequently throughout this manual. Filespec is an abbreviation of file specification, which is the full name of a file, its type, and the disk drive it resides on. Concurrent CP/M-86 uses a filespec to locate a file. File attributes and filetypes are also file descriptions that help Concurrent CP/M-86 locate the file you need and decide how to use it. These subjects are covered in greater detail in Section 2.

1.3.2 User Numbers

User numbers are one way Concurrent CP/M-86 helps you store files in an organized fashion. How you organize your files determines how quickly and easily you can find a file. Your Concurrent CP/M-86 system stores files on either floppy disks or fixed, nonremovable disks. By setting user numbers, you can divide either of these types of disks into multiple user areas that range in number from 0 through 15. You can store related files in specific user areas, which helps Concurrent CP/M-86 locate files faster and more easily. For instance, you can store all payroll information in User 1, correspondence in User 2, inventory files in User 3, your calendar file in User 4, and so on, through User 15.

To set user numbers, refer to the USER command in the Command Summary in Section 3.

1.4 One Last Word

As you read this manual, many words and phrases might be new to you. Appendix B, the User's Glossary, contains brief definitions of the computer terminology.

End of Section 1

1

2

3

4

Section 2

Let's Get Started

This section gives you examples of some of the Concurrent CP/M-86 features, tells you how to create files, and discusses many other concepts you need to use Concurrent CP/M-86 efficiently.

However, before you can use any programs with the system, you must load Concurrent CP/M-86 into the memory of the computer. The process of loading Concurrent CP/M-86 into memory is called boot loading, or booting, the system.

2.1 Booting Concurrent CP/M-86

Computers have slightly differing boot procedures. Some fixed disk systems might not require anything more than turning on the power; other systems might require insertion of one or more floppy disks into the drives prior to running Concurrent CP/M-86. Therefore, before you turn on the power switch on your system, read the owner's manual supplied by your system's manufacturer. It will tell you the proper procedure; boot the system accordingly.

In some cases, two floppy disks are necessary to boot Concurrent CP/M-86: the boot disk and another disk. The owner's manuals should tell you how many disks are required for your particular system, and which disks go into which drives.

While the system is booting into memory, a message might appear on your screen telling you that Concurrent CP/M-86 is loading into the computer. Many computers make a whirring or chirping noise while the system is loading.

When Concurrent CP/M-86 has finished loading into memory, a message with copyright information and the version number of your Concurrent CP/M-86 operating system appears on your console screen followed by the system prompt.

At this point, if you have included a start-up file, described in detail in Section 2.5.4, Concurrent CP/M-86 loads this file and executes the command line within as if you had typed it on the keyboard. Refer to Section 2.5.4 for an explanation of start-up files and ways to use them.

If you do not have a start-up file to run, you are ready to give Concurrent CP/M-86 commands. If you have a floppy disk system, you can remove the boot disk from the disk drive. The system disk that contains the programs, or utilities, discussed in this manual should be inserted in the system drive. The system disk is discussed later in this guide; check the manufacturer's instructions to determine which drive is the system drive.

2.2 What's on the Disk?

The programs you receive with Concurrent CP/M-86 help you create or edit files, protect files and disks, and combine and copy files. All programs and all files, including those shipped with Concurrent CP/M-86, those supplied by the system manufacturer, and those you create, are stored magnetically on disk. Each disk contains a directory that lists all the files located on that disk. This directory is divided into user number areas.

2.3 What Next?

The first thing you should do, whenever you get new software, is protect the disks that contain the programs by making a back-up copy of them. Humans make mistakes, and so do computers. Human or computer errors can destroy valuable programs and data files. You could accidentally erase a program by mistyping a command. An electrical power outage or electronic component problem could also damage the disks. So, always make a working copy of any program you purchase and put the original away. Then, if a program is accidentally erased, you can restore it from the original copy.

In addition, you should make frequent copies of programs and data files as you are developing them. Establish the habit of taking time at the end of each period of work to make back-up copies of that session's work. Backing up disks takes only a few minutes and it can save you hours—perhaps days or weeks—of work rebuilding files.

Any new disk that you purchase from a manufacturer must be formatted (prepared) before it can be used to make back-up copies of the system or to create new files. The programs you use to format disks and copy disks are hardware dependent. They are provided by the manufacturer of your computer. Read their instructions carefully and make the back-up copies of your Concurrent CP/M-86 disks before you continue reading this manual. With a back-up copy of your operating system, if you make a mistake, you can always use the original to make a back-up copy.

2.4 Drives and Devices

Concurrent CP/M-86 manages all the peripheral devices attached to your computer. Peripheral devices are pieces of hardware such as the disk drives, input devices such as keyboards, or terminals, and output devices such as printers. When you want to use a specific drive, or user number, or device, you must use a name the system understands. If you do not use specific designations, the system assumes that you want the default drive, user number, or device. The word default refers to the drive, user number, or other device that the system uses for retrieving or sending information unless you give specific instructions to do otherwise. The default drive, user number, and device are also called the current drive, current user number, and current device.

Concurrent CP/M-86 has one drive set apart from the others called the system drive. When you enter a command Concurrent CP/M-86 looks in several places for the program to handle that command. The first place Concurrent CP/M-86 searches is the default drive—the drive that is designated in the prompt by a letter. For example, if the prompt is A>, the default drive is A. Concurrent CP/M-86 always searches on the default drive, number, or device unless you provide the system with other information.

If you want to change the default drive, type the drive specification you want, followed by a colon, like this:

```
A>B:
```

This changes the default drive to the B drive. Unless you change the drive designation again, all the system prompt messages appear as

```
B>
```

Note: check your manufacturer's instructions to determine which drive assignments are valid on your computer.

With any user number above 0, Concurrent CP/M-86 reminds you which user number is currently in use by displaying the user number in front of the system prompt, like this:

```
3B>
```

If you need to change user numbers so you can access other files, enter the following command:

```
3B>user=7
```

Concurrent CP/M-86 changes the user number to 7. Until you change the user number again, the current user number is 7 and the system display looks like this:

```
7B>
```

The prompt tells you that Concurrent CP/M-86 accesses files in User Number 7 on the disk in drive B.

2.5 Types of Files

Concurrent CP/M-86's most important task is to access and maintain the files on your disks. Each file is a collection of information stored on a disk and every file must have a unique name so that Concurrent CP/M-86 can find it.

In general, there are two kinds of files: program files and data files. A program file is an executable file—a series of instructions the computer can follow. A data file is usually a collection of related information such as a list of names and addresses, the inventory of a store, or the text of a document.

There are also files which are collections of Concurrent CP/M-86 commands. These files have the filetype SUB and are executed with the SUBMIT command. Concurrent CP/M-86 supports a special type of file containing a single command line. As Concurrent CP/M-86 loads it looks for this start-up command file, described in Section 2.5.4.

2.5.1 Naming Files

Concurrent CP/M-86 identifies every file by its unique file specification. A file specification can have up to four parts: a drive specifier, a filename, a filetype, and a password.

A file name is simply a one- to eight-character name you assign to the file, such as

```
MYFILE2
```

When you make up a filename, choose a name that tells you something about the contents of the file. For example, if you have a list of vendor names for your business, you could name the file:

VENDORS

This filename gives you an idea of what is in the file.

As you begin to use Concurrent CP/M-86, you find that files fall naturally into families. To keep file families separated, Concurrent CP/M-86 allows you to add a family name, called a filetype, to the filename. For example, adding the NAM filetype to the filename VENDORS, as shown below, tells you the file contains a list of vendor names.

VENDORS.NAM

A filetype can be up to three characters long. Try to use three letters that tell something about the file's family. When you type a file specification for Concurrent CP/M-86 to read, separate the filetype from the filename with a period. When Concurrent CP/M-86 lists file specifications in response to a DIR command, it separates the filename from the filetype with blanks so that you can compare filetypes quickly.

Passwords protect directory labels or files from improper access, and the SET command in Section 3 describes their uses. A password can be any combination of letters or symbols (except * and ?), and up to eight characters long.

The following example shows a file specification that contains all the possible elements:

A:VENDORS.NAM;AMBER

In this example, A: is the drive specifier; VENDORS is the filename; .NAM is the filetype; and ;AMBER is the assigned password. Whenever you enter a filespec in a command line, remember that it is a complete entity and, therefore, no spaces are allowed between the various elements.

If you leave out the drive specifier, Concurrent CP/M-86 assumes you mean the default drive. If you leave out the password, the default password is assumed. If you leave out a filetype, Concurrent CP/M-86 might supply a filetype, depending on the context of its use, and in some cases a blank filetype is assigned.

2.5.2 Creating a File

There are many ways to create a file. The most common method of creating a file is with a text editing program or word processor. You can also create a file by copying an existing file to a new location, perhaps renaming it in the process, with the PIP command. Some programs, such as ASM-86™, create output files as they process input files.

2.5.3 Accessing Files

When you want to get a file you need to give the system all the information it needs to find that file. You have to specify what drive, or disk, the file is on, and the name of the file you want. If you just want to read the file called VENDORS.NAM, for instance, use the command, TYPE, like this:

```
A>TYPE VENDORS.NAM
```

If Concurrent CP/M-86 cannot find the file you requested on the A drive, it gives you the message:

```
File not found
```

If the file is on the disk in your B drive, enter the command shown below.

```
A>TYPE B:VENDORS.NAM
```

This command tells the system to search on the B drive. When the system finds the file, the contents scroll rapidly across your screen. If you press and hold down the control (CTRL) key along with the S key, the scrolling stops. To restart the scrolling, press the control key along with Q key. When you use the control key with alphabetic keys, you are using control character commands. Table 2-2 describes control character commands.

2.5.4 Start-up Files

You can execute a command automatically upon loading the system if you provide a start-up file for a particular virtual console. Each virtual console accepts a one-line command contained in a start-up file automatically loaded and executed when the system begins operation.

This command string can contain any Concurrent CP/M-86 command, such as DIR, USER, or VCMODE. If you want to execute more than a single command, specify the SUBMIT command with a filespec. Concurrent CP/M-86 then reads and executes the contents of the specified SUBMIT file, which can be as long and complex as you wish. Once this SUBMIT file finishes executing, the system prompt appears, and you can enter further instructions.

When you boot the system, Concurrent CP/M-86 looks for a file on the system drive with the name STARTUP.n where n is the number of a virtual console: STARTUP.0 controls virtual console 0, STARTUP.1 controls virtual console 1, and so forth. If you issue a SUBMIT command to direct execution from a SUBMIT file, follow the SUBMIT rules described in Section 3.17. You cannot SUBMIT the start-up file itself; it is loaded automatically upon start-up.

2.6 Computer Punctuation

As you were reading the last few paragraphs you probably noticed that the examples of filenames included periods, colons and semicolons. These punctuation marks are called delimiters because they separate, or mark off, the various parts of command lines, just as punctuation in written text does. Each delimiter has a separate and unique meaning to the system and, therefore, must never be used as a character in a file specification. Table 2-1 defines how each delimiter is used in Concurrent CP/M-86.

Table 2-1. Special Characters

<i>Character</i>	<i>Meaning</i>
< = , tab space, RETURN	The less than, equal, comma, tab space, and RETURN characters separate file references and other items in the command line.
:	The colon always follows the drive specifier.
.	A period separates the filetype from the filename.
;	A semicolon separates the filetype and password in a file specification.
* ?	The asterisk and question mark are wildcard characters. Refer to Section 2.7 for details.
< > and !	Angle brackets and the exclamation mark are reserved for future use.
[]	Square brackets isolate options from command keywords or file specifications.
()	Parentheses isolate lists of modifiers within options (see the SDIR command, Section 3).
/ \$	The slash and dollar sign specify options in the command line.
;	A semicolon at the beginning of a line indicates a comment that the command handler ignores.

2.7 Wildcard Characters

Wildcard characters are special characters that match certain specified items. In Concurrent CP/M-86, you can use two wildcard characters: the question mark and the asterisk. You can substitute the ? wildcard for any single character in a filename or filetype. You can use the * wildcard in one or both of the filename and filetype fields. The * wildcard matches the entire field or the remainder of the field in which you specify it. With wildcards, you can ask Concurrent CP/M-86 to look for one or more matching files.

Here are the rules for using wildcards:

- A ? wildcard matches any character in a name, including a space character.
- A * wildcard must be the last, or only, character in the filename or filetype. Concurrent CP/M-86 internally replaces a * with ? characters to the end of the filename or filetype.
- When the filename to match is shorter than eight characters, Concurrent CP/M-86 treats the name as if it ends with spaces.
- When the filetype to match is shorter than three characters, Concurrent CP/M-86 treats the filetype as if it ends with spaces.
- Wildcard characters can be used in the filename and filetype fields only; they cannot be used in the drive specifier or password fields.

The examples below show you how to use wildcards to match all or a portion of the following files:

A.CMD , AA.CMD , AAA.CMD , B.CMD , A.A86 , and B.A86

.	is treated as ?????????.???
?????????.???	matches all six names
*.CMD	is treated as ?????????.CMD
?????????.CMD	matches the first four names
?.CMD	matches A.CMD and B.CMD
?.*	is treated as ?.???
?.???	matches A.CMD, B.CMD, A.A86, and B.A86
A?.CMD	matches A.CMD and AA.CMD
A*.CMD	is treated as A????????.CMD
A?????????.CMD	matches A.CMD, AA.CMD, and AAA.CMD

2.8 Passwords

Concurrent CP/M-86 lets you protect your files with passwords. If more than one person uses your system, passwords are a way to protect your files from accidental damage by other users. Passwords also provide security for managers and systems personnel who want to limit access to particular files.

Any program, command, or data file can have individual password protection. The `PASSWORD` option of the `SET` command (see Section 3) lets you add this protection to files. Read the section on passwords for the `SET` command before you use the command to set password protection on any files.

Passwords can be up to eight characters long and can consist of any characters except those listed in Table 2-1.

2.9 Organizing Your Files

Under Concurrent CP/M-86 you can organize your files into groups, protect your files from accidental change, and specify how your files are displayed. These features are accomplished by assigning user numbers and attributes to files and recording them in the directory.

2.9.1. User Numbers

You can use user numbers to separate your files into 16 file groups. All files are identified by a user number that ranges from 0 to 15. Concurrent CP/M-86 assigns a user number to a file when the file is created. Generally, the file is assigned the current user number. You can change the current user number with the `USER` command described in Section 3.

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 created under user number 7. The exceptions to this are the `PIP` and `DIR [gn]` commands. `PIP` can copy a file from one user number and send the copy to another user number. The `DIR [gn]` option, where `n` is the desired user number, displays all files in user `n`. For further details, refer to the discussions of `DIR` and `PIP` in Section 3.

2.9.2 File Attributes

File attributes control how a file can be accessed. There are several file attributes you can set—the directory attribute, the Read-Write attribute, Read-Only attribute, the SYS attribute, and the Archive attribute. When you create a file, it is automatically marked with the DIR attribute. You can display the name of a file marked with the DIR attribute only if the file has the current user number. If you give a file the SYS attribute, it is not displayed in response to a DIR command; you must use SDIR or the DIR[S] command described in Section 3.

If you give the SYS attribute to a file for user number 0, you can read and execute that file from any user number. This feature gives you a convenient way to make your commonly-used programs available under any user number. However, note that a user 0 SYS file does not appear in response to a SDIR command unless 0 is the current user number. Also, SYS files for the current user number are not displayed unless you specify DIR[S].

The Read-Write attribute of a file gives you another way to protect your files. You can allow a file to be both read from and written to by using R/W (Read-Write), or you can set the file so that no editing can be done by setting R/O (Read-Only) on the file (see Section 3, SET Command). If you try to write data to a file with the R/O attribute set, Concurrent CP/M-86 returns a Read-Only error message. A file with the R/W attribute can be read from or written to at any time, unless the drive is set to Read-Only, or the disk is physically protected. See the manufacturer's notes on physically protecting removable disks. Usually, a 5-1/4 inch floppy disk is protected by covering the write-enable notch. On 8-inch floppy disks, uncovering the write-protect notch protects the disk.

The Archive attribute provides you with a convenient way of backing up your files. When you use the archive option in PIP, any time a file is changed, the attribute is turned off, allowing you to copy the file. If the file has not been modified, and the Archive attribute is turned on, the file is not copied when you make back-ups. Normally (without the archive option), PIP copies all files. For more details, refer to the PIP command in Section 3.

2.10 How Are Files Stored On Disks?

No matter what kind of computer you have, Concurrent CP/M-86 stores files the same way. The filename, filetype, user number, and attributes of each file are stored in a special area of the disk called the directory.

Concurrent CP/M-86 sets aside directory and storage space for a file only as the file grows. When you erase a file, Concurrent CP/M-86 reclaims the space that was allocated to the file. You do not need to tell Concurrent CP/M-86 how large you expect the file to become, because space is automatically allocated for each file as it is needed, and released for reallocation when the file is erased.

If you assign passwords or set time and date stamps on your files, the file requires additional directory space. Refer to the discussions of SET, and INITDIR in Section 3 for further details.

2.11 Making Mistakes

Everybody makes mistakes. If you accidentally type the wrong command or insert a disk incorrectly, the program running at the time should let you know. Normally, an error message appears on your console. There are many different types of error messages. Appendix A alphabetically lists the error messages returned by Concurrent CP/M-86. To give you an example of how this appendix is used, we can cause an error intentionally and then recover from it.

First mistype the DIR command in the following way:

```
C>DIRR
```

The system displays the following error message on the console:

```
CP/M Error: Can't Find Command  
Command = DIRR
```

If you do not understand the meaning of the error message, refer to the error message description and explanation, CP/M Error: Can't Find Command, in Appendix A for a more complete description. The message for this error tells you that you might have mistyped the command.

2.12 Control Character Commands

Some of the functions in Concurrent CP/M-86 are performed in response to single key commands in combination with the control key. They are called control character commands. They can tell Concurrent CP/M-86 to start and stop screen scrolling, or echo screen displays to the printer. Table 2-2 summarizes the control character commands.

Table 2-2. Concurrent CP/M-86 Control Character Commands

<i>Control Character</i>	<i>Meaning</i>
CTRL-C	Stops operation of the currently operating program and resets the disk system. If other programs have files open on a removable drive, Concurrent CP/M-86 tells you which drives are in use.
CTRL-E	Moves the cursor to the beginning of the following line without erasing any previous input and without adding any characters.
CTRL-H	Moves the cursor one character to the left and deletes the character.
CTRL-I	Moves the cursor to the next tab stop and automatically places a tab stop at each eighth column.
CTRL-J	Performs the same function as the RETURN key. It returns the cursor to the beginning of the next line and sends any commands to Concurrent CP/M-86.
CTRL-M	Same as CTRL-J; see CTRL-J.
CTRL-O	Ignores characters written to the screen until the next CTRL-O; these characters are lost. A CTRL-O lets you skip over sections of a long display.
CTRL-P	Tells Concurrent CP/M-86 to send screen output to the printer (if a printer is attached to the console) and to the screen. If the printer is already echoing the screen, CTRL-P halts the screen output to the printer.

Table 2-2. (continued)

<i>Control Character</i>	<i>Meaning</i>
CTRL-Q	Restarts the screen's display after you have used the CTRL-S command to stop it.
CTRL-R	Types a pound sign, #, at the current cursor location, moves the cursor to the same position on the next line, and retypes any partial command you entered.
CTRL-S	Starts screen scrolling.
CTRL-U	Discards all characters in a command line preceding the cursor, types a # at the cursor position, and moves the cursor to the next line.
CTRL-X	Discards all characters in a command line preceding the cursor and moves the cursor to the beginning of the current line.

2.13 Command Summary

Table 2-3 lists and briefly describes all the Concurrent CP/M-86 commands. Unless otherwise noted, Section 3 contains a detailed discussion of each command.

Table 2-3. Command Summary

<i>Command</i>	<i>Function</i>
ABORT	stops program execution on the virtual console that you specify in the command line.
ASM86	translates 8086 assembly language source programs into a machine-readable format. ASM-86 is documented in <i>Concurrent CP/M-86 Programmer's Utilities Guide</i> .
DATE	displays and/or sets the date and time.
DDT86	helps you examine assembly language programs and interactively correct programming errors. DDT-86 [™] is described in the <i>Concurrent CP/M-86 Programmer's Utilities Guide</i> .
DIR	lists the files on a specified disk or the default disk.
ED	lets you create and alter text, data, or program source files.
ERA	erases one or more files from a disk directory and releases the disk space occupied by the file.
ERAQ	erases one or more files from a disk directory, as ERA does, but ERAQ asks you to confirm the command for each file specified.
GENCCPM	lets you adjust XIOS parameters, allocate memory, and select Resident System Processes. GENCCPM is described in detail in the <i>Concurrent CP/M-86 System Guide</i> .
GENCMD	uses ASM-86 output to produce an executable command file. GENCMD is described in detail in the <i>Concurrent CP/M-86 Programmer's Reference Guide</i> .

Table 2-3. (continued)

<i>Command</i>	<i>Function</i>
HELP	displays information on how to use Concurrent CP/M-86 commands.
INITDIR	initializes a disk directory to allow date and time stamping on that disk.
PIP	combines and copies files.
PRINTER	shows the current default printer and lets you change it.
REN	lets you rename a file.
SDIR	displays a directory of system and nonsystem files and their attributes.
SET	lets you specify and alter certain file attributes.
SHOW	displays information about system resources such as the amount of usable space on a disk.
SUBMIT	sends a file of commands to Concurrent CP/M-86 for execution.
SYSTAT	displays the status of system resources such as memory and queues, and shows currently running processes. SYSTAT is described in the <i>Concurrent CP/M-86 Programmer's Reference Guide</i> .
TYPE	writes the contents of a text file to your screen.
USER	displays the current user number or changes one user number to another.
VCMODE	displays and/or sets the current background mode of the foreground virtual console.

End of Section 2

Section 3

Concurrent CP/M-86 Command Summary

This section shows you how to use Concurrent CP/M-86 commands and explains the parts of the command line. In addition, this section has a command summary, an alphabetically arranged reference section describing most of the Concurrent CP/M-86 commands. Each command is listed with examples and a short explanation of its operation.

3.1 Command Line Syntax

A command line always contains a command and might include a command tail. The command tail contains information the command uses, such as a file specification or options. Figure 3-1 uses the DIR command to illustrate the components of a command line.

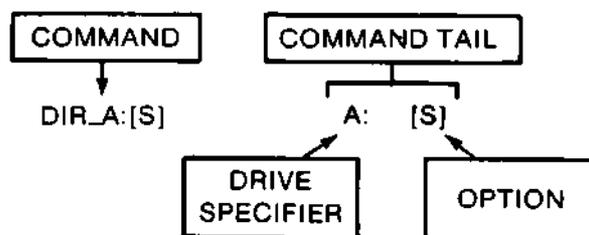


Figure 3-1. DIR Command and Command Tail

In the figure above, the DIR command tells the system to list files. In the command tail, the letter A followed by a colon is a drive specifier; it tells the DIR command to list the files on drive A. The letter S, enclosed in brackets, is an option that tells the DIR command to list only the system files. Thus, the DIR command in the figure lists the system files on drive A.

In Figure 3-1, the drive specifier is one of four components in a file specification. A file specification gives a program information the program requires to locate files on a disk. Table 3-1 lists and describes each of the components in a file specification. Figure 3-2 illustrates the general format of a file specification.

Table 3-1. File Specification Components

<i>Component</i>	<i>Description</i>
drive specifier	names the disk drive with a letter A through P. The drive specifier tells a program the name of the drive on which a file or group of files exists. The drive specifier is optional. However, if you specify a drive specifier, it must be followed by a colon as shown in Figures 3-1 and 3-2.
filename	a one- to eight-character name of a file or group of files.
filetype	a one- to three-character family name of a file or group of files. It follows the filename. Specifying a filetype for a file or group of files is optional. However, if you specify a filetype, separate the filetype from the filename with a period.
password	a one- to eight-character word that might be required to gain access to a file. If the password is required, separate it from the filename or filetype with a semicolon. Refer to the SET command in this section for details on passwords.

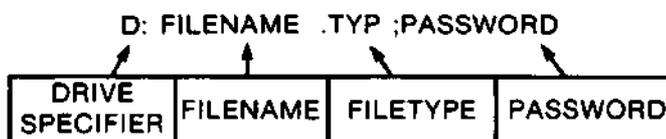


Figure 3-2. File Specification Format

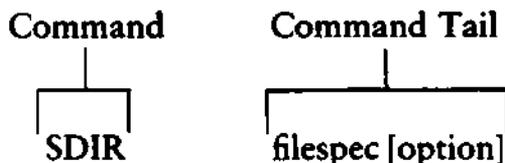
Figure 3-2 illustrates the general format of a file specification. The curly braces indicate optional items. The letters and symbols refer to the components described in Table 3-1. The following list shows the valid combinations of the components in a file specification.

- filename
- filename.typ
- filename;password
- filename.typ;password
- d:filename
- d:filename.typ
- d:filename;password
- d:filename.typ;password

A file specification indicates a particular file. If there are wildcard characters in the file specification, the filespec includes all files matching the wildcard characters. The directory of the disk named by the drive specifier contains the files described in the filespec. For example,

C:PAYROLL.LIB

The above example is a file specification that indicates drive C:, filename PAYROLL, and filetype LIB. In descriptions of command line syntax, the file specification is abbreviated to filespec as shown below.



Some Concurrent CP/M-86 commands accept wildcards in the filename and filetype of the command tail. For example,

B:PAY*.A??

is a file specification with drive specifier B:, filename PAY*, and filetype A??. This file specification might match several files in the directory, such as PAYROLL.A86 or PAYSCHED.ATT.

3.2 How Commands Are Described

The commands described in this section are in alphabetical order. The description begins with the command in upper-case. A syntax section follows, telling you one or more general forms to use in the command line. The text describes the operation of each command, noting special cases, suggesting programming techniques, and describing special conventions and considerations. The examples section provides examples of valid command lines. A list of error messages associated with each command is also included. For a detailed discussion of error messages, refer to Appendix A.

When you read the syntax section and specify commands, follow the guidelines listed below.

- Spell all commands as shown. However, unless otherwise noted, any combination of upper- and lower-case letters is acceptable.
- You can substitute any number for the variable *n*.
- The file specification components *d:*, filename, *.typ*, and password have the general meanings described in the previous section.
- You must include one or more space characters where a space is shown, unless otherwise specified. For example, the PIP options need not be separated by spaces.
- Items enclosed in curly braces are optional. You can enter a command without the optional items. The optional items add effects to your command line.
- An ellipsis (...) tells you that the previous item can be repeated any number of times.
- When you can enter one or more alternative items in the syntax line, a vertical bar | separates each alternative.
- All other punctuation, such as square brackets [], must be included in the command line.

Let's look at some examples of syntax. The Concurrent CP/M-86 SET command specifies password protection and time and date stamping of files. It also sets file and drive attributes such as the Read-Only (R/O), Read-Write (R/W), System (SYS), Directory (DIR), and user-definable attributes.

The syntax section of the SET command shows the following:

Syntax: SET d:[RW]
SET d:[RO]
SET filespec [SYS]
SET filespec [DIR]
SET [option = modifier]
SET d:[option = modifier]
SET filespec [option = modifier]
SET filespec [option]

Explanation: The SET command always requires an object and an action to perform upon the object. The object can be a drive, a file, or a group of files. The action can be any of a number of options described in this section. The options are always enclosed in square brackets. If you do not specify a drive or filename, SET assumes files are on the current drive. The system prompt tells you which drive is current; the alphabetic letter preceding the system prompt is the drive specifier. The current drive is also called the default drive.

The SET command includes options that effect entire drives and options that effect files or groups of files. SET options can be strung together and separated by commas inside the square brackets. Spaces before or after the brackets and equal sign are optional. You can use the SET command to set options on more than one file, but you must specify the option following each command, as shown in the following example:

```
A>SET B:INVOICE.LST [RO], RESPOND.TXT [RO]
```

The preceding SET command sets a Read-Only attribute to two files, INVOICE.LST and RESPOND.TXT.

The SET command lets you use wildcard characters in the filename and filetype fields of the file specification, as shown in the following example. With wildcard characters, you can set options for groups of files.

```
A>SET B:HELP.* [RO]
```

With the asterisk wildcard character in the filetype of the command line, the SET command sets the Read-Only attribute for all files with the filename of HELP on drive B.

There are several valid command lines that you can construct with the SET command. For instance:

```
A>SET B:[RW]
A>SET D:[RD]
A>SET B:INVOICE.LST [RW]
A>SET D:RESPOND.TXT [RD]
```

The above command lines show that you can set the Read-Write and Read-Only attributes for entire drives or individual files. For more details, refer to the SET command description in this section.

The ABORT Command

Syntax: ABORT filespec n

Explanation: The ABORT command stops the program that you specify in the file specification.

The variable n in the syntax line is the number of the console running the program. Substitute the console number for n in the command line. If you give the wrong file specification for the program name or the wrong console number, ABORT cannot stop the program, and the following message appears on the console screen:

```
Abort Failed
```

You can also stop a running program by typing a CTRL-C, but only if the program is running on the current virtual console and if the program has not disabled the CTRL-C function. Use ABORT to stop a program running on another console only if that program does not respond to CTRL-C.

Example: The following example illustrates a possible use of the ABORT command. The command is executed from the current virtual console and aborts a program running on another console. You can execute ABORT from any user number.

```
A>TYPE DOCUMENT.TXT <cr>
```

```
Dear Sir:
```

```
The company is plea_ <press key that switches screen>
```

```
A>ABORT TYPE 1 <cr>
```

```
A>
```

In this example, assume that the TYPE command was issued from virtual console 1, and that the console was switched at the end of the typed line, "The company is plea_..."

The DATE (Display and Set Time of Day) Command

Syntax: DATE {mm/dd/yy hh:mm:ss | P}

Explanation: The DATE command lets you examine or set the time and date on your system.

To specify a time and date, use the syntax shown above. Table 3-2 shows the abbreviations in the syntax line.

Table 3-2. Time and Date Specification Description

<i>Value</i>	<i>Meaning</i>
mm	specifies the month; a 2-digit number in the range 01 through 12.
dd	specifies the day; a 2-digit number in the range 01 through 31.
yy	specifies the year; a 2-digit value relative to 1900.
hh	specifies the hour; a 2-digit number in the range 00 through 24.
mm	specifies the minutes; a 2-digit number in the range 00 through 59.
ss	specifies the seconds; a 2-digit number in the range 00 through 59.

You represent a date by specifying the month with a 2-digit value in the range 01 through 12, the day with a 2-digit value in the range 0 through 31, and the year with a 2-digit value relative to 1900.

Time is represented as a 24-hour clock, with hour values from 00 to 11 for the morning, and 12 to 23 for the afternoon.

Be sure to use two digits for each field in the date (mm/dd/yy) and time (hh/mm/ss) specification. Specifying values like 12/9/82 does not work. If you specify only the date or the time, the DATE command does not set either.

Use the DATE command to obtain the current time and date. For example, the screen might appear as

```
Fri 12/10/82 14:15:23
```

in response to the DATE command. The command line to set time and date is shown below.

```
DATE mm/dd/yy hh:mm:ss
```

The following example sets the time and date for your system.

```
DATE 02/09/83 10:30:00
```

To let you set the time as accurately as possible, the command returns the message:

```
Strike key to set time
```

When the time that you give in the command tail occurs, press any key on the keyboard to set the time. DATE begins timing from that instant, and responds with a display in the form:

```
Wed 02/09/83 10:30:00
```

Use the command form,

```
DATE P
```

to print the time and date continuously on the screen. This display appears in the status line also.

You can stop the continuous display by pressing any key.

You need not set the time of day for proper operation of Concurrent CP/M-86.

Examples: A>DATE

This command writes the current time and date on the screen when you have set the time and date with the DATE command. If you have not set the date and time, the system returns a random time and date.

```
A>DATE 12/31/82 23:59:59
```

This command sets the current time and date to the last second of 1982.

The DIR (Directory) Command

Syntax: DIR {d:}
DIR {filespec}
DIR {filespec} [gn]

DIR[S] {d:}
DIR[S] {filespec}

Explanation: The DIR and DIR[S] commands display the names of files contained in the directory of a disk.

The DIR command lists the filenames of directory files that are in the current user area. Directory files are files that have the Directory (DIR) attribute set. The DIR[S] command displays a list of directory and system files that are in the current user area. System files are files that have the System (SYS) attribute set.

All command files in user area 0, regardless of their attribute, can be executed from any user number and any virtual console. However, you must be in user area 0 to display those files with the SDIR or DIR[S] commands. The DIR {filespec} [gn] command displays the files matching {filespec} in the user n directory space.

If a drive specifier and filename are omitted in the command line, the DIR command displays the names of all files with the DIR attribute on the disk in the default drive and current user number. When no drive specifier and filename are specified, the DIR[S] command displays all the files, including the system files, in the directory of the default drive and current user number.

If the drive specifier is included, but the filespec is omitted, DIR displays the names of all the DIR files in the current user area of the disk in the specified drive. DIR[S] adds the files with the SYS attribute to the listing.

Both forms of the command accept wildcards in the filename and file-type fields. When you use wildcard characters, all the filenames that satisfy the match are displayed on the screen. If no matches occur, or if no files are catalogued in the directory of the specified drive and user number, the DIR command displays the following message:

No File

If system files (SYS) that match the filespec reside on the specified drive and user area, DIR displays the message:

System Files Exist

Examples: A>DIR

This example displays all DIR files catalogued in user 0 on default drive A, while

A>DIR B:

displays all DIR files for user 0 on drive B.

A>DIR B:X.BAS

The above example displays the filename X.BAS if the file is on drive B.

4A>DIR *.BAS

This example displays all the DIR files with a .BAS filetype for user 4, drive A.

A>DIR A:X*.C?D

This example displays all the DIR files for user 0 on drive A with filenames that begin with the letter X, and have a filetype beginning with C and ending in D.

```
A>DIR[S]
```

The above example displays all the files for user 0, drive A. All files appear, even those having the SYS attribute.

```
3A>DIR[S] *.CMD
```

The asterisk wildcard character in the above filename tells the DIR[S] command to display all files, including system files, that have a filetype of CMD and exist on the default drive A in user 3.

The ED (Character File Editor) Commands

Syntax: ED input-filespec {d: | output-filespec}

Explanation: To do almost anything with a computer you need some way to enter and modify data before the computer can process the data. The programs most commonly used for this task are called editors. Concurrent CP/M-86's editor is called ED. Using ED, you can easily create or alter the file named in the input file specification.

ED is a versatile editor that contains many commands. Section 4 contains a tutorial of ED.

The ERA (Erase) Command

Syntax: ERA {d:} filespec

Explanation: The ERA command removes one or more files from the directory of a disk. Wildcard characters are accepted in the filename and filetype fields of 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 for the current user number are removed from the disk directory.

Command lines that take the form:

```
ERA {d:} *.*
```

require your acknowledgment, because they erase all the files on the disk in the specified drive.

```
Confirm delete all user files (Y/N)?
```

Type **y** to remove all files, and **n** to avoid erasing any files.

If you specify files in the file specification that Concurrent CP/M-86 cannot find in the directory, the following message appears on the console screen:

```
NO FILE
```

Examples: A>ERA X.A86

This command removes the file X.A86 from the disk in drive A.

```
A>ERA *.PRN
```

This command removes all files of filetype PRN from the disk in drive A.

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

All files on drive A with a filename that begins with MY are removed from the disk.

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

Confirm delete all user files (Y/N)?

If you type y, ERA deletes all files from the disk in drive B.

The ERAQ (Erase with Query) Command

Syntax: ERAQ filespec
ERAQ filespec[XFCB]

Explanation: The ERAQ command behaves exactly like the ERA command, with the addition of a query before each erasure.

The [XFCB] option of the ERAQ command erases only the extra information the system stores when you use passwords. If you do not use the SET command to set passwords you do not have this information on your disk and you do not need to use the [XFCB] option. If you have used passwords but do not need them any longer and use the SET command to turn off password protection, the [XFCB] option removes the extra information and space used by the information.

Example: 1A>ERAQ C:*.*CMD

```
C:ASM86      CMD ?y
C:FUNCTION   CMD ?n
C:SDIR       CMD ?y
C:DSKRESET   CMD ?y
1A>
```

In this example, ERAQ is instructed to delete all the files on the disk in drive C except C:FUNCTION.CMD.

The HELP Command

Syntax: HELP {topic} {subtopic1 subtopic2 ... subtopic8}

Explanation: The HELP command provides summarized information for all of the Concurrent CP/M-86 commands described in this manual. HELP with no command tail displays a list of all available topics. HELP with a topic in the command tail displays information about that topic, followed by any additional topics. HELP with a topic and a subtopic displays information about the specific subtopic.

After HELP displays the information for your specified topic, it displays the prompt HELP> on your screen. You can continue to specify topics for additional information, or simply press the RETURN key to return to the Concurrent CP/M-86 system prompt.

You can abbreviate the names of topics and subtopics. Usually one or two letters is enough to identify the topics. Typing a period stands for the current topic, so you could type .subtopic to obtain HELP for the current subtopic.

You should not specify a password for the HELP.HLP file. If you do specify a password for this file, you cannot read it.

Examples: A>HELP

When you enter the HELP command it produces a display of all the available HELP topics.

A>HELP PIP

The above command displays general information about the PIP command. It also displays any available subtopics.

A>HELP PIP OPTIONS

The above command includes the subtopic options. In response, HELP displays information about options associated with the PIP command.

A>HELP ED

The above command displays general information about the ED utility.

```
HELP>ED COMMANDS
```

The above example shows how to enter a topic and subtopic following the program's internal prompt, HELP>.

Creating Customized HELP Files

Syntax: HELP [EXTRACT]
HELP [CREATE]

Explanation: Concurrent CP/M-86 is distributed with two related HELP files; HELP.CMD and HELP.HLP. The HELP.CMD file is the command file that processes the text of the HELP.HLP file and displays it on the screen. The HELP.HLP file is a text file to which you can add customized information; however, you cannot directly edit the HELP.HLP file. You must use the HELP.CMD file to convert HELP.HLP to a file named HELP.DAT before you can add or edit your own text.

The HELP syntax shown above has the following options:

- The EXTRACT option accesses the file HELP.HLP on the default drive and creates a file called HELP.DAT on the default drive. You can now invoke ED or a word-processing program to edit and add your own text to the HELP.DAT file. EXTRACT can be abbreviated to E.
- The CREATE option accesses your edited HELP.DAT file on the default drive and builds a revised HELP.HLP file on the default drive. CREATE can be abbreviated to C.

You must add topics and subtopics to the HELP.DAT file in a specific format. The general format of a topic heading in the HELP.DAT file is

```
///nTopicname<cr>
```

The three backslashes are the topic delimiters and must begin in column one. In the format statement, *n* is a number from 1 through 9 that signifies the level of the topic. A main topic has a level number of 1, the first subtopic is level 2, and so forth to the ninth level. Topicname is the name of the topic, and allows a maximum of twelve characters. When you reach the end of the line, press the RETURN key.

To make your personal HELP file easier to use, we suggest that you place your topics in alphabetical order within their respective supertopics. Make sure that you indicate a topic level for all your entries.

Examples: `///1newutility<cr>`
`///2commands<cr>`
`///3options<cr>`
`///2examples<cr>`

The first example shows the format of a main topic line. The second example illustrates the first subtopic of new utility. Any topic name with a level of 2, or higher, is a subtopic within the main topic.

When you are executing the HELP.CMD file, you do not need to use all 12 characters. Only use those characters that you need to identify the topic clearly. When you are referencing a subtopic, you must type the topic name and the subtopic name. If you do not type both names, the HELP program does not know which main topic you are referencing. You can also enter a topic and subtopic following the HELP> prompt. For example,

```
HELP>ED COMM
```

This form of HELP displays information about commands in ED, the Concurrent CP/M-86 editor.

The INITDIR (Initialize Directory) Command

Syntax: INITDIR {d:}

Explanation: INITDIR initializes a disk directory to allow time and date stamping on that disk. Because time and date stamps require extra space on the directory, INITDIR reformats the directory—it does not reformat the disk. While the directory is reformatting, INITDIR saves all of your files then restores them when formatting is finished.

Time and date stamps require additional space in the directory. Therefore, if you set time and date stamps for files, more directory space is required than for files that do not have date and time stamps. Because of this space convention you must have at least 1/4 of the total directory space available. If the disk directory does not contain enough space to reformat the directory, INITDIR displays the following message:

```
ERROR: Not enough room in directory
```

To use INITDIR, enter a command such as

```
A>INITDIR C:
```

The system asks you to confirm the command by displaying the following message:

```
INITDIR WILL ACTIVATE TIME STAMPS FOR SPECIFIED DRIVE  
Do you want to re-format the directory on drive: C (Y/N)?
```

If the disk was previously formatted for time and date stamps, INITDIR displays the message:

```
Directory already re-formatted  
Do you want to recover time/date space (Y/N)?
```

If your directory is full, and you have enabled time and date stamps, you can use INITDIR to free up the directory space. If INITDIR finds time and date stamps in the directory, it displays the following message:

```
Do you want the existing time and date stamps cleared (Y/N)?
```

If the disk label is password protected, INITDIR displays the message:

```
Directory is Password Protected  
Password, please. >
```

If you do not enter the correct password, INITDIR terminates.

INITDIR is provided for disks which are already in use and might have been formatted on another CP/M system. To activate time and date stamping on new disks you must first format the disk according to your manufacturer's instructions.

The PIP (Peripheral Interchange Program—Copy File) Command

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

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 and can combine two or more files into one file. PIP also copies a character file from disk to the printer or other auxiliary logical output device. Either console input or input from some other logical input device is used to create files on disk. PIP can transfer data from a logical input device to a logical output device, hence the name Peripheral Interchange Program.

Note: before you use PIP to copy from one disk to another, be sure that both disks are the same format. If the disks are formatted differently from each other, reformat them according to the manufacturer's instructions.

Single File Copy

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

Explanation: The first syntax form above is the simplest way to copy a file. PIP looks for the file named by source-filespec on the default or optionally specified drive and copies the file to the drive specified by d:, giving 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]. However, you can specify several options for the source file. "Using Options with PIP" contains a description of the PIP options.

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.

Before you start PIP, use the `SHOW` command to make sure that you have enough free space on the destination disk to hold the file 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.

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`, `R/W`, `R/O`) are transferred with the files. If you want to copy a file with the `SYS` attribute you must add `[r]` after the source filename.

If the existing destination file is set to Read-Only (`R/O`), PIP asks you if you want to delete it. Answer `Y` or `N`. Use the `W` option to write over Read-Only files without being queried.

You can include PIP options following each source name (see "PIP Options"). 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 and they can be packed together or separated by spaces. There are many PIP options; you can verify that a file was copied correctly, allow PIP to read a file with the system (`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 upper-case, and so on.

Examples: `A>PIP B:=oldfile.dat`

`A>PIP B:oldfile.dat=A:`

Both 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. For the destination file, specify a drive

or user number that differs from the source file. When you do not specify a user number, drive, filename, or filetype that differs from the source, PIP copies the file to the same drive and user number but overwrites the original file.

```
A>PIP B:newfile.dat=A:oldfile.dat
```

This 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 includes a filename and filetype on both sides of the command line.

```
A>PIP newfile.dat=oldfile.dat
```

Using this long form of PIP, you can copy a file from one drive and 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 B:PROGRAM.BAK[G1]=A:PROGRAM.DAT
```

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

```
B>PIP program2.dat[G3]=A:program1.dat[EV]
```

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

Multiple File Copy and Making Back-ups

Syntax: PIP d:{{Gn}} = {d:}wildcard-filespec{{options}}

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

If you want to copy files only when they have recent modifications, use the archive option of PIP. When the [A] option is enabled, files with the archive bit set are not copied. (See Table 3-3, PIP options). When an archive file is successfully copied, PIP sets the archive bit for the original copy of the file, and Concurrent CP/M-86 turns the bit off whenever the file is modified. This allows you to copy only those files which you have changed since the last time you used PIP with the archive option. This feature is very useful when you want to back up only those files which have changed since you last made back-up copies.

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

This command causes PIP to copy all the files on drive A with the filetype **CMD** to drive B.

A>PIP B:=A:*.DAT[A]

This command uses the archiving option of PIP. It causes PIP to copy all the files (on drive A with the filetype **.DAT**) that have been modified since the last time files were archived.

A>PIP B:=A:*. *

This command causes PIP to copy all the files on drive A to drive B. You can use this command to make a back-up copy of your distribution disk. Note, however, that this command does not copy the Concurrent CP/M-86 system from the system tracks.

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

All files of any filetype, in which the filename consists of PROG or begins with the letters PROG, are copied from drive A to drive B with this command.

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

This PIP command copies all the files with a filetype of BAK in user number 0 on drive A to user number 1 on drive B. 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 utility.)

Combining Files

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

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 in 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 location of the end of the file. All of the PIP options force a character transfer except the following:

Gn O R V W

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

When PIP is concatenating files it 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 the [O] option if you are concatenating machine object code files. The [O] option causes PIP to ignore the embedded CTRL-Z (end-of-file) characters which are normally used to indicate the end of character files.

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

The three files named FILE1, FILE2, and FILE3 are joined from left to right and copied to NEWFILE.***. NEWFILE.*** is renamed to NEWFILE when the copy operation is complete. All source and destination files are on the disk in the default drive A.

`A>PIP B:X.A86 = Y.A86 , B:Z.A86`

The file Y.A86 on drive A is joined with Z.A86 from drive B and placed in the temporary file X.*** on drive B. The file X.*** is renamed to X.A86 on drive B after PIP successfully copies and concatenates the files.

Copy Files to and from Auxiliary Devices

Syntax: `PIP dest-filespec[{{Gn}}|dev = source-filespec|dev{{o}}`

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 ASCII characters. Each peripheral device has 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. Strike any key to abort a copy operation that uses a logical device in the source or destination.

The logical device names are as follows:

- **CON:** The console; the physical device assigned to CON:. When used as a source, CON: is usually the keyboard; when used as a destination, CON: is usually the screen.
- **LST:** The destination device assigned to LST:, usually the printer.

There are three device names that have special meaning:

- NUL: A source device used for paper tape readers that produces 40 nulls (hexadecimal zeros).
- EOF: A source device that produces a single CTRL-Z (the Concurrent CP/M-86 end-of-file mark).
- PRN: The printer device with tab expansion to every eighth column, line numbers, and page ejects every 60th line.

Examples: `A>PIP B:FUNFILE.SUE=CON:`

If CON: is assigned to input, whatever you type at the console is written to the file FUNFILE.SUE on drive B. End the keyboard input by typing a CTRL-Z.

`A>PIP LST:=CON:`

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

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

The file DRAFT.TXT on drive B is written to the printer device. Any tab characters are expanded to the nearest column that is a multiple of 8.

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

The command above causes PIP to write the file DRAFT.TXT to the list device. It automatically expands the tabs, adds line numbers, and ejects pages after 60 lines.

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

Characters are read from the console input device first, generally the keyboard, and sent directly to the printer. 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.

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 prompts with an asterisk on your screen when it is ready to accept input command lines. You can type any valid command line described under previous PIP formats after the asterisk prompt appears.

To stop the PIP program, press the RETURN key following the asterisk prompt. The empty command line tells PIP to discontinue operation and return to the Concurrent CP/M-86 system prompt.

Note: this form of PIP allows changing disks between commands. Before changing disks, you should switch to another virtual console and type a CTRL-C. If there are any open files on the system, you will be notified of which disks are in use. If nothing happens or your drive is not listed, you can change the disk. This verifies that no other programs have open files on the disk you are about to change. Also, this allows the system to check for changes in density or number of sides on the new disk when it is placed in the drive.

Examples:

```
A>PIP
*NEWFILE=FILE1,FILE2,FILE3
*APROG.COM=BPROG.COM
*A:=B:X.AGG
*B:=*,*
*
```

This 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 line is included in the command tail. PIP is not loaded into memory for each command.

Using Options With PIP

Options enable you to process your source file in special ways. Among other options, you can expand tab characters, translate from upper- to lower-case, extract portions of your text, and verify that the copy is correct.

The 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 [G*n*] option after a source or 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 3-3. PIP Options

<i>Option</i>	<i>Meaning</i>
A	To back up only the files that have been modified since the last back-up, use PIP with an ambiguous filename and the Archive option. PIP with the [A] option copies only the files that have been modified.
Dn	Deletes any characters past column n. Use this option for a source file with lines that are 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.
C	Asks you to confirm each transfer.
E	Echoes transfer at console. When this option 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	Filters form-feeds. When this option follows a source filename, PIP removes all form-feeds embedded in the data in the source file.
Gn	Go to user number n. When this option 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.
H	Hex data transfer. PIP checks all data for proper Intel® hexadecimal file formats. The console displays error messages when errors occur.
I	Ignores :00 records in the transfer of Intel hexadecimal format file. The I option automatically sets the H option.
L	Translates upper-case alphabets in the source file to lower-case in the destination file. This option follows the source device or filename.
N	Adds line numbers to the destination file. When this option follows the source filename, PIP adds a line number to each line copied. The first line starts with the number 1 and successive line numbers are incremented 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 option is also set, PIP expands the tab.

Table 3-3. (continued)

Option	Meaning
O	Object file transfer for machine code (noncharacter and therefore nonprintable) files. PIP ignores any CTRL-Z ends-of-file during concatenation and transfer. Use this option if you are combining object code files.
Pn	Sets page length. The variable n specifies the number of lines per page. When this option 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	Quits copying from the source device after the string s. When used with the S option, the Qs option can extract a portion of a source file. The string argument must be terminated by CTRL-Z. This option is valid only when PIP is in Multiple Command Mode.
R	Reads system (SYS) files. Normally, PIP ignores files marked with the system attribute in the disk directory. But when this option follows a source filename, PIP copies system files, including their attributes, to the destination.
Ss	Starts copying from the source device at the string s. The string argument must be terminated by CTRL-Z. When used with the Q option, the Ss option can extract a portion of a source file. Both start and quit strings are included in the destination. This option is valid only when PIP is in Multiple Command Mode as described in Section 3.11.5.
Tn	Expands tabs. When this option 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 3-3. (continued)

Option	Meaning
U	Translates lower-case alphabetic characters in the source file to upper-case in the destination file. This option follows the source device or filename.
V	Verifies that data has been copied correctly. PIP compares the destination to the source data to ensure that the data has been written correctly. The destination must be a disk file.
W	Writes over files with R/O (Read-Only) attribute. Normally, if a PIP command tail includes an existing R/O file as a destination, PIP sends a query to the console to make sure you want to write over the existing file. When this option follows a source name, PIP overwrites the R/O file without a console exchange. If the command tail contains multiple source files, this option need follow only the last file in the list.
Z	Zeros the parity bit. When this option 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: `A>PIP NEWPROG.A86=CODE.A86[L],DATA.A86[U]`

This command constructs the file NEWPROG.A86 on drive A by joining the two files CODE.A86 and DATA.A86 from drive A. During the copy operation, CODE.A86 is translated to lower-case, while DATA.A86 is translated to upper-case.

`A>PIP CON:=WIDEFIL.E.A86[DB0]`

This command writes the character file WIDEFIL.E.A86 from drive A to the console device, but deletes all characters following the 80th column position.

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

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 LST:=B:LONGPAGE.TXT[FP65]
```

This 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.

```
B>PIP LST:=PROGRAM.A86[NTBU]
```

This command writes the file PROGRAM.A86 from drive B to the printer device. The N option tells PIP to number each line. The T8 option expands tabs to every eighth column. The U option translates lower-case letters to upper-case as the file is printed.

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

This command abstracts 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. When you specify a character string that contains upper- and lower-case, you must specify the command in response to the asterisk PIP command prompt, rather than from the system prompt, which is A> in this example. PIP returns the * prompt when PIP is in multiple command mode. Refer to the "Multiple Command Mode" section for details.

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

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

The PRINTER Command

Syntax: PRINTER {n}

Explanation: The PRINTER command displays or sets the printer used by a particular console. Several consoles can share the same printer, but only one process can use the printer at a time. Concurrent CP/M-86 expects the first printer assigned to the system to be printer number 0. The second printer is printer 1, and so on. The printer number is specified by n. When the PRINTER command includes an n, PRINTER sets the list device to printer number n. When you enter PRINTER without the n option, the system returns the number of the printer currently assigned to your console.

Examples:

```
A>PRINTER
Printer Number = 1
A>PRINTER 2
Printer Number = 2
```

Note: your system might or might not support more than one printer. Read the instructions from your computer's manufacturer to determine how many printers your particular computer can use.

The REN (Rename) Command

Syntax: REN {d:}newname{.typ} = oldname{.typ}

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

The filename *oldname* identifies an existing file on the disk. The filename *newname* is not in the directory of the disk. The REN command changes the file named by *oldname* to the new filename given in the command line.

REN changes only the name of the file—it does not make a copy 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 specification as a part of the newname. If both file specifications name a drive, it must be the same drive. If the file given by *oldname* does not exist, REN displays the following message on the screen:

```
No such file to rename
```

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

```
Not renamed: newname already exists, delete (Y/N)?
```

Examples: A>REN NEWASM.A86=OLDFILE.A86

The file OLDFILE.A86 changes to NEWASM.A86 on drive A.

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

The file Y.PLI changes to X.PAS on drive A.

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

The file OLDLIST changes to NEWLIST on drive B. The second drive name, B:, is implied by the first one, and is unnecessary in this command line. The preceding command line has the same effect as

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

The SDIR Command

Syntax: SDIR
SDIR d:
SDIR d:[option]
SDIR filespec
SDIR filespec [option]
SDIR filespec,filespec [option]
SDIR [option]
SDIR [option = modifier]
SDIR [option]d:
SDIR [option = modifier] d:
SDIR [option] filespec,filespec
SDIR [option = modifier] filespec,filespec

Explanation: The SDIR utility is equipped with all of the options needed to display Concurrent CP/M-86 files in a variety of ways. SDIR can search for files on any or all drives, in any or all user areas.

SDIR supports only global options—those which modify the entire command line. Formal global options are only allowed after the command name on the command line. SDIR allows the option list to occur anywhere on the command line; however, only one option list is allowed.

Options must be enclosed in square brackets. The options can be used individually or strung together, separated by commas. Only one or two letters are needed to identify the option unambiguously. The right bracket is needed only if the option is followed by a drive specifier or file specification. SDIR with no specified options displays files in the default user area on the default drive.

SDIR searches on single file specifications or on combinations of up to 10 file specifications. Refer to 3.1 for details on file specifications. SDIR ignores passwords. SDIR allows the asterisk and question mark wildcard characters in the filename and filetype fields. When you specify only the drive specifier without a filename or filetype, SDIR interprets the command line as if you specified the following command line:

```
A>SDIR d:*,*,*
```

SDIR has two formats: the default or full format and the size format.

SDIR Full Format

The default or full format shows the name of the file, the size of the file in number of records and in number of kilobytes, and the attributes of the file. If there is a directory label on the drive, SDIR shows the password protection mode and the time stamps. If there is no directory label, SDIR displays two file entries on a line, omitting the password and time stamp columns. The display is alphabetically sorted.

The following is an example of an SDIR full display. Because Drive M has no directory label, SDIR displays two files per line.

```
GE>sdir m:
```

```
Directory For Drive M: User 6
```

Name	Bytes	Recs	Attributes	Name	Bytes	Recs	Attributes	
FRONT	PRN	5K	34	Dir RW	FRONT	TEX	3K 24	Dir RW
TOC	PRN	6K	46	Dir RW	TOC	TEX	4K 29	Dir RW

```
Total Bytes      = 18K  Total Records = 133  Files Found = 4
Total 1k Blocks = 18   Used/Max Dir Entries For Drive M: 4/ 64
```

SDIR displays the Read-Only or Read-Write, the SYS or DIR, the Archive attribute, and other user-defined (F1, F2, F3, F4) attributes of a file. SDIR displays SYS if the System attribute of the file is on, and DIR if it is off. SDIR displays R/O if the Read-Only attribute is on, and R/W if it is off. SDIR displays the number 1, 2, 3, or 4 corresponding to the number of any user attributes that are on.

The full format displays two measures of file size. The size of the file in kilobytes is the total amount of disk space allocated to the file by the operating system. The number of records is the actual file length in 128 byte units. Depending on the size of a block on the disk, the operating system in general allocates more storage than is needed by the file. The disk block size is the minimum allocation unit.

SDIR Size Format

The second format is the size format. SDIR [SIZE] displays the filename and the file size in kilobytes.

Both the full format and the size format follow their display with two lines of totals. The first line displays the total number of kilobytes, the total number of records, and the total number of files listed. The second line displays the total number of 1k blocks needed to store the listed files. The number of 1k blocks shows the amount of storage needed to store the files on a single-density disk or on any drive that has a block size of 1 kilobyte. The second line also shows the number of directory entries used per number of directory entries available on the entire drive. These totals are suppressed if only one file is found.

SDIR Options

The following table explains each of the SDIR options.

Table 3-4. SDIR Options

<i>Command</i>	<i>Result</i>
SDIR	displays all files on the default drive, in the default user area, in full format, sorted alphabetically. This command is the default display and is equivalent to SDIR = [Full].
SDIR [SYS]	displays only the files that have the SYS attribute on.
SDIR [RD]	displays only the files that have Read-Only attribute on.
SDIR [DIR]	displays only the files that have the SYS attribute off.
SDIR [RW]	displays only the files that are set to Read-Write.

Table 3-4. (continued)

<i>Command</i>	<i>Result</i>
SDIR [XFCB]	displays all the files that contain the additional information required for passwords. Refer to the SET command for a discussion of passwords.
SDIR [NONXFCB]	displays those files without Extended File Control Blocks.
SDIR [USER=n]	displays system and directory files under the user number specified by n.
SDIR [USER=ALL]	displays files under all the user numbers for the default drive.
SDIR [USER=(0,1,...,15)]	displays files under the user numbers specified.
SDIR [DRIVE=d]	displays files on the drive specified by d. The drive specified must exist. DISK is also acceptable in place of DRIVE in all the DRIVE options.
SDIR [DRIVE=ALL]	displays files on all of the logged-in drives. A logged-in drive is a valid existing drive that has been accessed since the last disk reset.
SDIR [DRIVE=(A,B,C,...,P)]	displays files on the drives specified. The specified drives must exist.
SDIR [FULL]	displays the files in full format. If no directory label exists for the drive, or the NONXFCB option was specified, the columns for time and date stamps are omitted, and two files are displayed per line. This is the default display.

Table 3-4. (continued)

<i>Command</i>	<i>Result</i>
SDIR [LENGTH=n]	displays n lines of filenames before inserting a table heading. n must be in the range between 5 and 65536.
SDIR [SIZE]	displays the disk space in kilobytes allotted to the files on the default or specified drive.
SDIR [FF]	The form-feed option is used with the CTRL-P character to make hard copies of directories. It sends an initial form-feed to the printer. If the LENGTH option is also specified, a form-feed is issued every n lines.
SDIR [MESSAGE]	The message option is used when SDIR is searching for files on more than one drive and/or user area. Normally, SDIR does not print the names of the drives and users it is searching. With this option, SDIR displays the names of the specified drives and user areas and any files residing there. If no files exist in the specified locations, SDIR displays the File not found message.
SDIR [NOSORT]	SDIR normally sorts files alphabetically. SDIR [NOSORT] displays the files in the order it finds them in the directory.
SDIR [EXCLUDE]	SDIR with the EXCLUDE option displays the files that do not match the files specified in the command line.
SDIR [ATTRIBUTES]	replaces the "Arcv" information with "A1234" information, indicating which file attributes are turned on.

Examples: The following examples illustrate some of the uses of the SDIR command. The following command line instructs SDIR to list all the system files with the filetypes PLI, CMD, and A86 that are on the currently logged-in drives in all user areas.

```
A>SDIR [user=all,drive=all,sys] *.PLI *.CMD *.A86
```

The following example instructs SDIR to display the filename TESTFILE.BOB if it is found on any logged-in drive or user area.

```
A>SDIR [drive=all user=all] TESTFILE.BOB
```

The example below instructs SDIR to list each Read-Write file that resides on drive D and to specify the size of the file in kilobytes. Note that specifying only the drive specifier, d:, with the SDIR command is equivalent to specifying d:.* with the SDIR command.

```
A>SDIR [size,rw] D:
```

The following example lists all the PRL files on drive D that require additional information in the directory because they have passwords set:

```
A>SDIR [xpcb] D:*.CMD
```

The following SDIR command lists all the files on the default drive and user area that do not have a filetype of CMD:

```
A>SDIR [exclude] *.CMD
```

The SET Command

Syntax: SET [options]
SET d: [options]
SET filespec [options]

Explanation: The SET command initiates password protection and time stamping of files and establishes the drive and file attributes such as READ-ONLY, SYS, or user-defined attributes. SET also lets you label a disk and password-protect the label. Before enabling the time stamping option, you must first format the disk directory. (See "The INITDIR Command" for information on disk directory formatting).

Assign a Label to the Drive

Syntax: SET {d:} [NAME = labelname.typ]

Explanation: This SET command assigns a name to the specified drive's directory, or to the default drive's directory if no drive is listed. You can create a directory label with an eight character name and three character type, similar to filename and filetype. For example, the command:

```
A>SET [NAME=DISK100]
```

labels the disk on the default drive, DISK100. The default label name is LABEL.

Enable/Disable Password Protection

Syntax: SET [PROTECT = ON]
SET [PROTECT = OFF]

Explanation: The first SET command turns on password protection for all the files on the disk. The second SET command disables password protection for all the files on the disk. Password protection must be turned on before you assign passwords to individual files or labels.

Assign Password to the Label

Syntax: SET [PASSWORD = password]
 SET [PASSWORD = <RET>

Explanation: The first syntax form of this SET command assigns the password to the disk label. The second form of the command removes the password protection from the label.

If a label has no password, any user who has access to the SET program can set other attributes to the disk (which might make the disk inaccessible to you). However, if you assign a password to the label, you must also supply the password to set any of the functions controlled by the label. SET always prompts for the password if the label is password protected. You can determine if a disk is password protected by using SHOW to display the label (See "The SHOW Command").

When you assign or change any password on a drive, you must enter the commands in a specific order. Password protection must be switched on before you can assign a password. Set the password like this:

```
D>SET [PASS=XYZ]
```

Once the drive has the protection turned on and the password set, it prompts you for the password whenever you access that drive or label.

If you use password protection on your disks, be sure to record the password. If you forget the password, you lose access to any protected files.

Assign Passwords to Files

Syntax: SET filespec [PASSWORD = password]

Explanation: This command sets the password for a file to the password that you specify with the PASSWORD option. Passwords can be up to eight characters long. Lower-case letters are translated to upper-case. For example

```
A>SET MYFILE.TEX [PASSWORD=MYFIL]
```

assigns the password MYFIL to the file MYFILE.TEX.

You can use the asterisk and question mark wildcard characters in the filename and filetype fields of the filespec. SET assigns the specified password for all files that match the wildcard filespec. For information on wildcard characters, refer to Section 2.7.

Set Password Protection Mode

Syntax: SET filespec [PROTECT = READ]
SET filespec [PROTECT = WRITE]
SET filespec [PROTECT = DELETE]
SET filespec [PROTECT = NONE]

Explanation: You can assign one of four password protection modes—READ, WRITE, DELETE, and NONE—to your files. Table 3-5 describes each mode.

Table 3-5. Password Protection Modes

<i>Mode</i>	<i>Protection</i>
READ	The password is required for reading, copying, writing, deleting, or renaming the file.
WRITE	The password is required for writing, deleting, or renaming the file. You do not need a password to read the file.
DELETE	The password is only required for deleting or renaming the file. You do not need a password to read or modify the file.
NONE	No password exists for this field. If a password exists, this modifier is used to delete the password.

For example

```
B>SET *.TEX [PASSWORD=SECRET, PROTECT=WRITE]
```

assigns the password SECRET to all the TEX files on drive B. Each TEX file is given a WRITE protect mode to prevent unauthorized editing.

Assign a Default Password

Syntax: SET [DEFAULT=password]

Explanation: This command assigns a default password that the system uses during a given session. The system uses the default password to access password protected files if you do not specify a password. The system lets you access the file if the default password matches the password assigned to the file. The default password must be set for each virtual console.

For example,

```
A>SET [DEFAULT=dd]
```

instructs the system to use dd as a password if you do not enter a password for a password protected file.

SET Time Stamp Options on Disk

Syntax: SET [CREATE = ON]
SET [ACCESS = ON]
SET [UPDATE = ON]

Explanation: These SET commands allow you to keep a record of the times and dates you created, accessed, or updated your files. However, before you enable time stamping on your files, you must first run INITDIR to format the disk directory for time and date stamps. (See "The INITDIR Command.")

The CREATE option allows you to record the creation time of a file. It turns on the CREATE time stamp on the default or specified disk. You must turn the option on before you create the file.

The ACCESS option turns on the ACCESS time stamps on the disk in the default or specified drive. ACCESS and CREATE are mutually exclusive—only one can be in effect at any given time. If you turn the ACCESS time stamp on a disk that has CREATE active, the CREATE time stamp is turned off.

The UPDATE option turns on UPDATE time stamps on the disk in the default drive. UPDATE records the last time you modified, or updated, your file.

Although there are three kinds of time and date stamps, only two can be associated with a file at one time. Also, time and date stamps cannot be selectively assigned to individual files. Either all files on a disk are stamped, or none at all.

If you set both UPDATE and CREATE, both stamps are changed when you edit a file. This occurs because ED does not update the original file but rather creates a new version with the same name.

For example, if you have ACCESS time stamps turned on and you enter the command:

B>SDIR [FULL]

the system displays time and date stamps in the following format:

Directory for Drive B:

Name	Bytes	Recs	Attributes	Prot	Update	Access
ONE	.TEX	9K	71	Dir RW	None	08/03/81 10:56
THREE	.TEX	12K	95	Dir RW	None	08/05/81 15:45
TWO	.TEX	10K	76	Dir RW	None	08/10/81 09:30

The ACCESS time stamps displayed show the time the file was last displayed or edited. Note that displaying a filename in a directory listing does not constitute an access and is not recorded.

If you turn on CREATE and UPDATE, the display includes CREATE and UPDATE in the columns at the right, as shown below.

Name	Bytes	Recs	Attributes	Prot	Update	Create
GENLED	.DAT	109K	873	Dir RW	None	08/05/81 14:01 08/01/81 09:36
RECEIPTS	.DAT	59K	475	Dir RW	None	08/08/81 12:11 08/01/81 09:40
INVOICES	.DAT	76K	608	Dir RW	None	08/08/81 08:46 08/01/81 10:51

Set Drive Attribute

Syntax: SET {d:} [RO]
SET {d:} [RW]

Explanation: These SET commands set the specified drive to Read-Only, or Read-Write. If your drive is password protected and protection is on, the system asks you for your password. If a drive is set to Read-Only, PIP cannot copy a file to it, ERA cannot delete a file from it, and REN cannot rename a file on it. You cannot perform any operation that requires writing to the disk. When the specified drive is set to Read-Write, you can read from or write to the disk in that drive.

Additional Set Examples

Examples: A>SET *.CMD [SYS,RO,PASS=123,PROT=READ]
A>SET *.CMD [RW,PROTECT=NONE,DIR]

The first command affords the most protection for all the CMD files on drive A. The password protection is set to READ, therefore you cannot even read one of the CMD files without entering the password, 123, unless the default password is set to 123.

The second command reverses the protection and access attributes of the CMD file. Anyone can read and write the files; no password is required. However, the system prompts you for the password before it allows you to turn off the password protection.

Set FILE Attribute Bits

Syntax: SET filespec [Fn = ON/OFF]

Explanation: The SET utility allows you to set file attribute bits for specified files. For more information on file attribute bits, see the *Concurrent CP/M-86 Operating System Programmer's Reference Guide*. To set the F1 attribute on all files starting with the letters "WS", for example, you would type

```
SET WS*.* [F1 = ON]
```

The SHOW Command

Syntax: SHOW
 SHOW [SPACE]
 SHOW [DRIVES]
 SHOW [USERS]
 SHOW [LABEL]
 SHOW [HELP]

 SHOW [d:]
 SHOW [d:SPACE]
 SHOW [d:DRIVE]
 SHOW [d:USERS]
 SHOW [d:LABEL]

Explanation: You can also string commands together:

 SHOW [USERS,SPACE,LABEL]

The SHOW utility provides information regarding the status of the specified disk. It displays the drive, the Read-Only or Read-Write mode for that drive, and the remaining space in kilobytes for all logged in drives in the system. The SHOW [SPACE] display is the same as the SHOW display. SHOW [HELP] displays a list of the SHOW options. SHOW with the optional drive specifier displays the SHOW information for the specified drive only, as shown in the following example. If you use an option with SHOW, you must enclose it in square brackets: SHOW [option].

Examples: A>SHOW B:
 B: RW, Space: 9,488K

The SHOW [DRIVES] command displays the drive characteristics of logged-in drives on the system, or for a specified drive. The following is an example of the SHOW [DRIVES] display:

```

A: Drive Characteristics
3,600: 128 Byte Record Capacity
450: Kilobyte Drive Capacity
96: 32 Byte Directory Entries
96: Checked Directory Entries
128: Records / Directory Entry
16: Records / Block
48: Sectors / Track
2: Reserved Tracks
    
```

The SHOW [USERS] command displays the current user number and all user areas on the drive that have files assigned to them, as shown in:

```

A>SHOW [USERS]
Active User : 1
Active Files: 0 2 3 4
A:* of files: 29 17 1 5
    
```

The SHOW [LABEL] command returns a display of the optional directory label, if it has been created. The directory label is an entry within the directory and contains information that describes special attributes of the drive to the operating system. For example, the label tells whether time stamping and password protection are turned on for that drive.

You can assign a label that helps you identify the data stored on the disk. You also can assign a password to the label to prevent unauthorized access to the SET label functions that turn on or turn off time stamping and password protection.

You create a directory label when you use a SET command, regardless of password protection or time stamping for a drive. See the "SET Command" for details. The following example illustrates the SHOW [LABEL] display:

```

A>SHOW [LABEL]
Label for drive B:

Directory  Passwds  Stamp  Stamp  Label Created  Label Updated
Label      Rend     Create Update
-----
TDMSDISK.  on       on      on      07/04/81 10:30  07/08/81 09:30
    
```

The first column, Directory Label, displays the name assigned to that drive directory by a SET [NAME = TOMSDISK] command. The second column, Passwds Req'd, shows that password protection has been turned on for that drive with a SET [PROTECT = ON] command. If password protection is turned off, all password protection on the drive is deactivated, which means that you can access, update, and even delete any password protected files on the drive even though you do not know the password. This option is normally set on. It should only be set off if the password to an important file is forgotten. If password protection is used, be sure to assign a password to the label itself so that an unauthorized person cannot turn off password protection. It is important to remember the Directory Label password.

As mentioned above, each file can have up to two time stamps. The first of these time stamps can be either the creation date and time for the file, or the time and date of the last access to the file (access is defined as reading from or writing to the file). The fourth column of the SHOW [LABEL] command displays both the type of stamp and whether or not it is on. In the example above, creation time stamps are given to new files as shown by the Stamp Create column heading. Creation time stamps can be turned on for the drive with a SET [CREATE = ON] command.

If access time stamps are preferred, they can be turned on for the drive with a SET [ACCESS = ON] command. In this case, the fourth column heading in the SHOW [LABEL] display is Stamp Access instead of Stamp Create. It is best to decide whether you want creation date stamping for files or access date stamping for files, set the appropriate mode, and leave it that way. Otherwise, once changed from ACCESS back to CREATE, ACCESS dates for files might be in the Stamp Create field.

The fifth column displays the status of the second time stamp field, the update time stamp. Update time stamps display the time and date of the last update to a file, that is, the last time someone wrote to the file. In the display above, update time stamps have been turned on with a SET [UPDATE = ON] command.

In addition to showing the password protection and the active time stamps on a drive, SHOW [LABEL] also displays the time and date that the label was created and last updated.

The SUBMIT Command

Syntax: SUBMIT filespec { parameters... }

Explanation: 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 deliver all the commands at once with the SUBMIT utility.

To do this, create a file and enter your commands in this file, one per line, as you normally would enter them at the system prompt. The SUBMIT file is identified by its filename and must have a filetype of SUB. When you say SUBMIT filespec, the SUBMIT utility reads the file and interprets it as a sequence of Concurrent CP/M-86 commands.

SUBMIT looks for the SUB file in the default user number directory first. If it cannot find the file, SUBMIT looks on the system drive in user 0 for a SUB file set to SYS. Thus, you can set up system submit files just like your system CMD files.

The file of type SUB can contain any valid Concurrent CP/M-86 commands. A command line cannot exceed 125 characters.

If you want, you can include SUBMIT parameters filled in by values included in the command tail. SUBMIT parameters take the form of a dollar sign, \$, followed by a number in the range 0 through 9:

\$0 \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9

You can put these parameters anywhere in the command lines in your SUB file.

The SUBMIT utility reads the command line and substitutes the items you entered in the command tail for the parameters included in the file of type SUB. When the substitutions are complete, SUBMIT sends the file to Concurrent CP/M-86 line-by-line, as if you entered each command.

Each item in the command tail is a sequence of alphabetic, numeric, or special characters. The items are separated by one or more blanks.

The first word in the command tail takes the place of \$1 in the SUB file, the second word replaces \$2, and so forth, through the last item in the command tail. The filename of the SUB file replaces any \$0 parameters in the SUB file. Usually, however, \$0 is not used.

If you type fewer items in the command tail than parameters in the SUB file, remaining parameters are replaced by empty (null) strings.

If you type more items in the command tail than parameters in the SUB file, the items remaining in the command tail are ignored.

Command processing stops after the system reads the last line of the SUB file. Pressing a CTRL-C stops the system from processing the commands in a SUB file.

SUB files can contain nested commands. To include a nested SUB file within another SUB file, give a command of the form:

```
$INCLUDE FILE.SUB
```

where FILE.SUB is another SUBMIT file.

To include an actual dollar sign in your file of type SUB, type two dollar signs, \$\$. The SUBMIT utility replaces them with a single dollar sign when it substitutes a command tail item for a \$ parameter in the SUB file. This is especially useful for commands such as ASM86 that use the \$ as option delimiters (see example below).

You can include passwords in your SUBMIT file. You cannot embed a CTRL-P in the file to turn on the printer.

\$GLOBAL and \$LOCAL Variables:

There are three commands you can use in SUBMIT files that can alter processing at the system level after the SUBMIT file finishes executing. You can write a file that resets the user number with the USER utility, specifies a new printer with the PRINTER facility, or changes the default disk with the d: convention.

If you only want these changes to affect program execution within the SUBMIT file, preface these commands with the \$LOCAL option. When the SUBMIT file is finished executing, the user number, printer number, and/or drive number return to their former values. This is the default operation of SUBMIT.

If you want these changes to remain valid after the execution of the SUBMIT file, preface these commands with the \$GLOBAL option. When the SUBMIT file is finished executing, the user number, printer number, and/or drive number retain the values specified in the SUBMIT file.

Examples: `A>SUBMIT SUBFILE`

Assume the file SUBFILE.SUB is on the disk in drive A, and that it contains the following lines:

```
DIR*.CMD
ASM86 X $$SB
PIP LST:= X.PRNT8D80]
```

The SUBMIT command shown above sends the sequence of commands contained in SUBFILE.SUB to Concurrent CP/M-86 for processing. Concurrent CP/M-86 first performs the DIR command and then assembles X.A86. When ASM86 finishes, the PIP command line is executed.

```
A>SUBMIT B:ASMCMD X  B D80  SZ  ←  These command tail
                               $1 $2 $3 $4 ←  items are assigned
                                               to these SUB file $n
                                               parameters.
```

Assume that ASMCMD.SUB is present on drive B and that it contains the following commands:

```
ERA $1,BAK
ASM86 $1 $$4
PIP LST:= $1.PRNT$2 $3 $5]

PIP LST:= $1.PRNT$2 $3 $5]
```

The SUBMIT utility reads this file and substitutes the items in the command tail for the parameters in the SUB file as follows:

```
ERA X.BAK
ASMBG X $SZ
PIP LST. = X, PRNLTB D80]
```

Concurrent CP/M-86 executes these commands from top to bottom.

The following program limits the action of the drive specification and the USER command to within the SUBMIT file. When the SUBMIT file is finished executing, the user number and drive number return to their former values.

```
$LOCAL
B:
USER 2
PAYROLL.CMD
```

The following program lets the action of the drive specification and the USER command remain after the end of the SUBMIT file execution. When the SUBMIT file is finished executing, the user number and drive number retain the values specified in the SUBMIT file. A program such as this is useful as a SYS file located in user 0 on the system drive.

```
$GLOBAL
B:
USER 2
DIR
```

The TYPE (Display File) Command

Syntax: TYPE filespec

Explanation: The TYPE command displays the contents of a text file on your console screen.

If the file specified by filespec is not present on the default or specified disk, TYPE displays the following message on your screen:

```
NO FILE
```

To list the file on both printer and screen, press a CTRL-P before you enter the TYPE command line. To stop echoing keyboard input at the printer, press a second CTRL-P.

TYPE returns the following error message if you give it a filespec that contains wildcards; TYPE can type only one file at a time:

```
No wildcards allowed
```

Examples: A>TYPE MYPROG.A86

This command displays the contents of the file MYPROG.A86 on your screen.

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

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

The USER (Display and Set User Number) Command

Syntax: USER {n}

Explanation: The USER command displays and changes the current user number. The disk directory can be divided into distinct areas according to a user number.

The default user number is 0. When Concurrent CP/M-86 starts, it assumes that 0 is the current user number. With the exception of User 0, Concurrent CP/M-86 displays the current user number in the system prompt; the current user number precedes the drive specifier:

3A>

This number can be in the range 0 to 15. To display the current user number, use the command form:

USER

To change the current user number, use the command form:

USER n

where n is a number in the range 0 through 15.

Give a command of the form:

SHOW [USER]

to obtain a list of user numbers that have files associated with them.

Examples: `A>USER`

The above command displays the current user number on your screen, shown below:

```
USER NUMBER=0
```

To change the current user number to 3, type the following command:

```
A>USER=3
```

The system responds with:

```
3A>
```

The VCMODE (Set Virtual Console Background Mode) Command

Syntax: VCMODE
VCMODE DYNAMIC
VCMODE BUFFERED
VCMODE SIZE=N

Explanation: The VCMODE command lets you specify background operating modes for the virtual consoles.

When a Concurrent CP/M-86 virtual console is switched-out, or in the background, it operates in one of two possible modes: the Dynamic mode or the Buffered mode.

If a virtual console is in Dynamic mode and you switch it into the background, data normally output to the screen fills a space in memory reserved for such data. If there is enough data output to more than fill this area, the oldest data is lost as the new data is written in, just as when new lines appear on the bottom of the screen and old ones disappear from the top. In this case, when you switch back to this console some data might be lost on the screen.

If a console is in Buffered mode and switched-out, data output from a running program goes into a buffer file on disk rather than to the screen storage area in memory. Concurrent CP/M-86 writes these buffer files onto the temporary drive. When you switch in this console, the system fetches the data stored in the buffer file on the temporary disk and writes it onto the screen. These files disappear after their data appears on the screen. If the temporary disk becomes full and cannot store any more output from the program, the program hibernates until you switch in to the console.

If you want to suppress the screen output from a switched-in buffered console, press CTRL-O. You will not see any characters until you press CTRL-O again. You can use the control character CTRL-S to stop data scrolling past your screen and CTRL-Q to restart the data.

You can specify the maximum size of the disk buffer file with the VCMODE command. If a buffer file fills up, Concurrent CP/M-86 suspends that console's program operation; no data is lost. Operation begins when you switch this console in again. You cannot specify a buffer size for a console set to Dynamic mode since no buffer is used in Dynamic mode. If you try to use a buffer, Concurrent CP/M-86 merely reports that the console is set to Dynamic Mode, and nothing else happens.

Try out these examples and you will quickly become familiar with how Concurrent CP/M-86 manages multiple virtual consoles.

Examples: If you ask for help, or type anything other than a VCMODE command, VCMODE explains itself briefly:

```
A>VCMODE HELP
      VCMODE EXAMPLES

vcmode          (show background mode)
vcmode dynamic  (sets background mode)
vcmode buffered (sets background mode)
vcmode size = 5 (sets buffered mode max file size in)
                 (kilobytes; legal range is 1 to 8191)
vcmode help     (prints this message)
```

Typing VCMODE by itself returns a line telling you what the background mode for your current console is, as shown in the following example:

```
A>VCMODE

Background Mode For Virtual Console    0 is Dynamic
```

When you enter the command VCMODE DYNAMIC, the VCMODE program returns a line confirming the command:

```
A>VCMODE DYNAMIC

Background Mode For Virtual Console    0 set to Dynamic
```

Entering VCMODE BUFFERED returns a line confirming the command and tells you how large the maximum file size is, as shown below:

A>VCMODE BUFFERED

Background Mode For Virtual Console 0 set to Buffered
Maximum file size = 12K

If you try to specify the maximum file size when the current console is in Dynamic mode, you get the following:

A>VCMODE SIZE=5

Background Mode For Virtual Console 0 set to Dynamic

Set it to Buffered Mode first:

A>VCMODE BUFFERED

Background Mode For Virtual Console 0 set to Buffered
Maximum file size = 12K

Then, specify the maximum file size:

A>VCMODE SIZE=5

Background Mode For Virtual Console 0 set to Buffered
Maximum file size = 5k

End of Section 3

Section 4

ED, the Concurrent CP/M-86 Context Editor

4.1 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 file. Concurrent CP/M-86's editor is named ED. Using ED, you can easily create and alter Concurrent CP/M-86 text files.

The correct command format for invoking the Concurrent CP/M-86 editor is given in Section 4.2, "Starting ED." After starting ED, you issue commands that transfer text from a disk file to memory for editing. Section 4.3, "ED Operation," details this operation and describes the basic text transfer commands that allow you to easily enter and exit the editor.

Section 4.4, "Basic Editing Commands," details the commands that edit a file. Section 4.5, "Combining ED Commands," describes how to combine the basic commands to edit more efficiently. Although you can edit any file with the basic ED commands, ED provides several more commands that perform more complicated editing functions, as described in Section 4.6, "Advanced ED Commands."

During an editing session, ED can return two types of error messages. Section 4.7, "ED Error Messages," lists these messages and provides examples that indicate how to recover from common editing error conditions.

4.2 Starting ED

Syntax:

ED input-filespec output-filespec

To start ED, enter its name after the Concurrent CP/M-86 prompt. The command ED must be followed by a file specification that contains no wildcard characters, such as the one shown in the following example:

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

The file specification, MYFILE.TEX in the above example, specifies a file to be edited or created. The file specification can be preceded by a drive specification, but a drive specification is unnecessary if the file to be edited is on your default drive. Optionally, the file specification can be followed by a drive specification, 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, ED automatically creates a new file and displays the following message:

```
NEW FILE
: *
```

Note: before starting an editing session, use the STAT command to determine how much free space is left on your disk. Make sure that the unused portion of your disk is at least as large as the file you are editing—larger if you plan to add characters to the file.

4.3 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 back-up 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 file and an output file, ED does not rename the input file to type .BAK; therefore, the input file can be Read-Only.

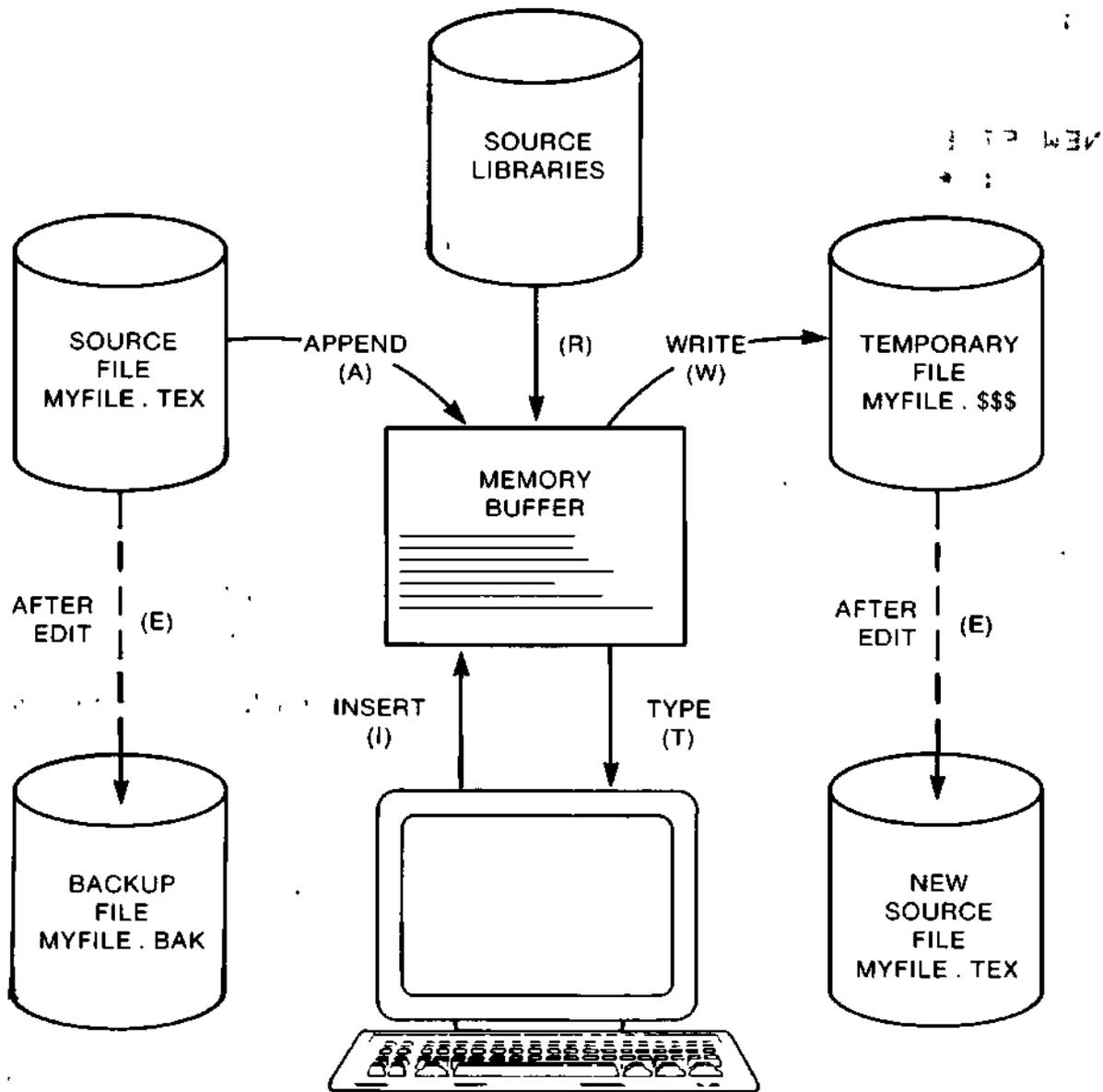


Figure 4-1. Overall ED Operation

In the preceding figure, 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 4-1. Text Transfer Commands

<i>Command</i>	<i>Result</i>
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.

4.3.1 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.

The V (Verify Line Numbers) Command

The V command turns the line number display in front of each line of text on or off. The V command also displays the free bytes and total size of the memory buffer. The forms of the V command are

V, -V, 0V

Initially, the line number display is on. Use `-V` to turn it off. If the memory buffer is empty, or if the current line is at the end of the memory buffer, ED represents the line number as five blanks. The 0V command prints the memory buffer statistics in the form:

free/total

where free is the number of free bytes in the memory buffer, and total is the size of the memory buffer. For example, if you have a total of 48,253 bytes in the memory buffer and 46,652 of them are free, the OV command displays this information as shown below:

```
46652/48253
```

The A (Append) Command

The A command appends (copies) lines from an existing source file into the memory buffer. The form of the A command is

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.

When 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.

4.3.2 ED Exit

You can use the W (Write) command and 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.

The W (Write) Command

The W command writes lines from the buffer to the new file. The form of the W command 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 `OW` command is a convenient way of making room in the memory buffer for more lines from the source file. If the buffer is full, you can use the `OW` command to write half the contents of the memory buffer to the new file. You can use the `#W` command to write the entire contents of the buffer to the new file. Then you can use the `OA` command to read in more lines from the source file.

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

The 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 `$$$` file. If a `.BAK` file exists, ED deletes it, then renames the original file with the `.BAK` filetype. Finally, ED renames the `$$$` 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 back-up 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 back-up 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.

4.4 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 CP's position is always right before the character you are pointing at.

Note: throughout the discussion of ED, the symbol, ^, indicates the position of the character pointer.

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 be 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 4-2. Basic Editing Commands

<i>Command</i>	<i>Action</i>
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 CTRL-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 affect the position of the character pointer in the memory buffer. Later examples in Section 4.5, "Combining ED Commands," illustrate how the commands appear at the screen. For these sections, however, the symbol ^ in command examples represents the character pointer, which you must imagine in the memory buffer.

4.4.1 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.

The B (Beginning/Bottom) Command

The B command moves the CP to the beginning or bottom of the memory buffer. The forms of the B command are

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.

The C (Character) Command

The C command moves the CP forward or backward the specified number of characters. The forms of the C command are

nC, -nC

when 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 is pointing to the beginning of the line, the command 30C moves the CP to the next line:

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

The 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 two lines forward through the memory buffer and positions the character pointer at the beginning of the line.

```

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

The 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. For example, the command -2 moves the CP back two lines.

```

Emily Dickinson said,<cr><lf>
^"I find ecstasy in living -<cr><lf>
the mere sense of living<cr><lf>
is joy enough."<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 a n command of 1 and moves the CP down to the next line and prints it, as shown below.

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

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

4.4.2 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.

The T (Type) Command

The T command types a specified number of lines from the CP at the screen. The forms of the T command are

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<cr><lf>
is joy enough."<cr><lf>
```

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 press a CTRL-S to stop the display, then press any key when you are ready to continue scrolling. Press a CTRL-C to abort long type-outs.

4.4.3 Deleting Characters

The D (Delete) Command

The D command deletes a specified number of characters and takes 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 two characters.

The 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:

```
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 OK 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.

4.4.4 Inserting Characters into the Memory Buffer

The I (Insert) Command

To insert characters into the memory buffer from the screen, use the I command. If you enter the command in upper-case, ED automatically converts the string to upper-case. The I command takes the forms:

```
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 RETURN 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 insert or command mode, you can use Concurrent CP/M-86 line-editing control characters to edit your input. Note, however, that you cannot use CTRL-E in insert mode. The following table lists these control characters.

Table 4-3. Concurrent CP/M-86 Line Editing Controls

<i>Command</i>	<i>Result</i>
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.
CTRL-X	Delete the entire line currently being typed. Same as CTRL-U.
Backspace	Remove the last character.

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

The 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 Concurrent CP/M-86'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:

```
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>
```

4.4.5 Replacing Characters

The S (Substitute) Command

The S command searches the memory buffer for the specified string, but when it finds it, automatically substitutes a new string for the search string. Whenever you enter a command in upper-case, ED automatically converts the string to upper-case. 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:

```
sEmily 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 upper-case 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.

4.5 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 Concurrent CP/M-86's line-editing controls to manipulate ED command strings.

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

- The combined command line must not exceed Concurrent CP/M-86'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, S, J, X, and R 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.

4.5.1 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: *
```

4.5.2 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: *OLT
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: *BCT
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: *
```

4.5.3 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: *15C5DOLT
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 OLT command string as shown below. Remember that the insert string must be terminated by a CTRL-Z.

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

4.6 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.

4.6.1 Moving the CP and Displaying Text

The 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.

The 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 line number command takes the form:

`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.

The :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 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:

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

4.6.2 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.

The F (Find) Command

The F command performs the simplest find function. The F command takes the form:

```
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: *Zfliving
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: *Zfliving^Zolt
3: *the mere sense of living
```

It makes a difference whether you enter the F command in upper- or lower-case. If you enter F, ED internally translates the argument string to upper-case. If you specify *f*, ED looks for an exact match. For example, FConcurrent CP/M-86 searches for CONCURRENT CP/M-86 but fConcurrent CP/M-86 searches for Concurrent CP/M-86, and cannot find CONCURRENT CP/M-86.

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.

The 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 N command takes the form:

```
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. The case of the N command has the same effect on an N command as it does on 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, but 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 OA 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.

The 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 J command takes the form:

```
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: *OIt
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 upper-case J command letter is specified, ED looks for upper-case search and delete-to strings and inserts an upper-case insert string.

The J command is especially useful when revising comments in assembly language source code, as shown in the following example:

```

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 containing strings can exceed 100 characters. When you finish your command, press the carriage return key to send the command to ED.

The 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-BdiLiving^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, and not M.

To abort a macro command, press a CTRL-C.

The Z (Sleep) Command

Use the Z command to make the editor pause between operations. The pauses give you a chance to review what you have done. The form of the Z command is,

nZ

where n is the number of seconds to wait before proceeding to the next instruction.

Generally, the Z command has no real effect unless you use it with a macro command. The example below shows you how you can use the Z command to cause a ten second pause each time ED finds the word TEXT in a file.

```
A>*mflliving`Z0tt10z
```

4.6.3 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.

The X (Transfer) Command

The X command takes the forms,

nX
nXfilespec^Z

where n is the number of lines from the CP towards the bottom of the buffer that are to be transferred to a file. Therefore, n must always be a positive number. The nX command with no file specified creates a temporary file named X\$\$\$\$\$.LIB. This file is erased when you terminate the edit session. The nX command with a file specified creates a file of the specified name. If no filetype is specified, .LIB is assumed. This file is saved when you terminate the edit session. 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.

The 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 where ED types the entire file to verify the read.

```
1: #41
1: : *R^ZB#T
1: "I find ecstasy in living -
2: the mere sense of living
3: is joy enough."
4: Emily Dickinson said,
1: *
```

4.6.4 Saving or Abandoning Changes: ED Exit

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

The 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 H command takes the form:

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.

The 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 O command takes the form:

O

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

```
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.

The Q (Quit) Command

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

```
Q
```

followed by a carriage return.

When you enter a Q command, ED prompts you to verify that you want to abandon the changes:

```
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 Concurrent CP/M-86.

Note: you can enter a CTRL-C to return control immediately to Concurrent CP/M-86. This does not give ED a chance to close the source or new files, but it prevents ED from deleting any temporary files.

4.7 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 Concurrent CP/M-86 error message if ED cannot read or write to the specified file. An ED error message takes the form:

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 4-4. ED Error Symbols

<i>Symbol</i>	<i>Meaning</i>
#	Search failure. ED cannot find the string specified in a 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 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.

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

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

```
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 file error messages ED returns when it cannot read or write a file.

Table 4-5. ED File Error Messages

<i>Message</i>	<i>Meaning</i>
<code>Bdos Err on d: R/O Function NNN File: FILENAME.TYP</code>	Disk d: has Read-Only attribute. This occurs if a different disk has been inserted in the drive since the last time the system was booted.
<code>** FILE IS READ ONLY **</code>	The file specified in the command to invoke ED has the R/O attribute. ED can read the file so that the user can examine it, but ED cannot change a Read-Only file.
<code>File Not Available</code>	The file specified in the command is open on another virtual console, and ED cannot get access to it.

End of Section 4



Appendix A

Concurrent CP/M-86 Error Messages

The first section of this appendix is a quick reference table listing Concurrent CP/M-86 error messages in alphabetical order, with explanations and suggestions for remedying the problem. The second section discusses several common error handling procedures, and provides more detail about causes of certain types of errors.

A.1 Concurrent CP/M-86 Error Messages

Error messages come from several different sources:

- Each utility supplied with Concurrent CP/M-86 has its own set of error messages.
- Concurrent CP/M-86 also displays error messages when errors occur during execution of file system operations and when certain error conditions occur.

Table A-1 lists these error messages in alphabetical order, with explanations and suggestions for resolving the problem.

Table A-1. Concurrent CP/M-86 Error Messages

<i>Message</i>	<i>Meaning</i>
When absolutely nothing happens:	<p>First, wait. Some programs take a long time to finish. Some errors, such as Disk I/O errors, cause Concurrent CP/M-86 to retry the operation many times. With a little experience, you will know when a command is taking too long.</p> <p>If you have waited long enough (1-5 minutes maximum for most Concurrent CP/M-86 utilities) without obtaining a response, your console might be confused and not functioning correctly. This can happen when two programs try to use the same part of the computer simultaneously. For example, if a program on a virtual console is printing, or is running a word processor that was printing, then another program requiring the printer must stop and wait for the first program to give up the printer.</p> <p>Another more subtle conflict for resources can occur when you call DIR. If you run DIR on one console and stop it with a CTRL-S, then switch to another console and run DIR again, the second console might hang, waiting for the first DIR to finish. Simply switch to the first console running DIR and either let it finish by entering a CTRL-Q, or abort it with a CTRL-C.</p>
Abort Failed	<p>ABORT error. You specified the wrong program name, or the wrong console. Ensure that you specify the correct program name and the correct console and reenter the command line. If ABORT still does not work, go to the console where the program you want to abort is running and press a CTRL-C.</p>
CP/M Bad Sector	<p>REN, SET error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
BREAK "x" AT c	<p>ED error. "x" is one of the symbols described below and c is the command letter being executed when the error occurred.</p> <ul style="list-style-type: none"> * Search failure. ED cannot find the string specified in a 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. D The file specified in a R command could not be found. > Buffer full. ED cannot put any more characters in the memory buffer, or the string specified in an F, N, or S command is too long. E Command aborted. A keystroke at the console aborted command execution. F Disk or directory full. This error is followed by either the disk full or directory full message. Refer to the recovery procedures listed under these messages.
Cannot Edit Wildcard Filename	<p>ED error. You requested a wildcard (*) filename when invoking ED. ED does not support wildcard filenames as sources. Give a specific filename and reenter the ED command line.</p>
Could not protect or reset drive.	<p>SET error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
CP/M Error: Bad File Spec	<p>TMP error. You requested an invalid file specification. Ensure that you are typing the correct file specification and reenter the command line.</p>
CP/M Error: Can't Find Command	<p>TMP error. Concurrent CP/M-86 cannot find your command file. You might have mistyped the command. If not, the command file is not on the default drive or the system disk, or is not in the default user number, usually 0 (see the USER utility description for details of setting user numbers.) In this case, you must switch to the correct user number and specify the drive when you enter the command: for example, e:command.</p> <p>This error might also occur when the command file is on the system disk in user 0 but does not have the system attribute. Use the SET command to assign the command file the system attribute before you reenter the command line. Refer to SET [HELP] or to "The SET Command," for more information on setting file attributes.</p> <p>There is a discussion of the search path Concurrent CP/M-86 uses to find files in Section A.2.1 of this appendix.</p>
CP/M Error: CLI Abort	<p>TMP error. The Command Line Interpreter (CLI) was stopped, typically by a CTRL-C. If you did not intend to abort the command, you must reenter the command line.</p>
CP/M Error: Read Error	<p>TMP error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
CP/M Error: Invalid Drive	<p>TMP error. The disk you specified in the command line is not supported by your system. Concurrent CP/M-86 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.</p>
CP/M Error: Load Error	<p>TMP error. This message indicates a corrupted command file. If the command file is a program that you have written, refer to Section A.2.2 of this appendix for a discussion of recovery procedures for corrupted command files.</p> <p>If this message occurs any other time, and you have a back-up copy of the program you were running, erase the program file and replace it from your back-up files. If the error recurs, contact your technical support group of the manufacturer of your system for assistance in locating the cause of the error, or consult with the vendor you purchased your system from. Refer to Section A.2.6 of this appendix, on reporting logic errors, for a list of the information you will need to provide them to diagnose your problem.</p>
CP/M Error: Not Enough Memory	<p>TMP error. There is not enough memory available to load the command file you specified. If you are running one or more large programs on another console, wait until they are finished and reenter the command line. If there were no other programs running at the time this error occurred, you might not have enough memory resident in your particular system to run the program, or you might have a corrupted file. Refer to Section A.2.4 of this appendix for more information. This error can also occur when the command file is corrupted. Refer to Section A.2.2 of this appendix, on Corrupted Files, for more information.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
CP/M Error: PD Table Full	TMP error. You have reached the maximum number of processes possible with your particular hardware. Either wait until one or more processes are finished, or use ABORT or CTRL-C to terminate one or more processes and free up space. This error might affect more than just the program that received the error message. Either check the results of the other programs carefully or rerun everything. If this error occurs frequently, refer to Section 2.10, File Open Modes.
CP/M Error: RSP Command Queue Full	TMP error. The command you specified is tied up right now, probably on another console. Retry later, or check your other consoles to locate the command and either wait for it to finish, or abort the command with CTRL-C or ABORT. If this error occurs frequently, refer to Section A.2.5 of this appendix.
CP/M Error on d: Close Checksum Error	BDOS error. A file cannot be closed properly because the directory Concurrent CP/M-86 finds on the disk does not match the one logged into memory. This error usually occurs when the disk has not been reset. Press CTRL-C. If the error persists, ensure that no other processes are running and reboot the system. If the error recurs, it might mean that the application you are running is incompatible with Concurrent CP/M-86. Refer to Section A.2.5 of this appendix for further information.
CP/M Error on d: Disk I/O	BDOS error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.

Table A-1. (continued)

Message	Meaning
CP/M Error on d: Disk reset denied	<p>BDOS error. A program tried to reset a drive with an open file. Check your other consoles to locate the program with the open file and either wait for it to finish or abort the program with a CTRL-C or the ABORT command.</p>
CP/M Error on d: File Already Exists	<p>BDOS error. You tried to create or rename a file when there is already a file of that filename and filetype on the specified drive and user number. Either select a different name or use the ERA command to delete the existing file before reentering the command line.</p>
CP/M Error on d: File Currently Open	<p>BDOS error. A process tried to delete, rename, or modify an attribute of a file currently opened by another process. Either wait until the other process is finished and the file closed, or use the ABORT command or CTRL-C to stop the process.</p> <p>Some applications programs produce this error if you run them on two consoles simultaneously, as Concurrent CP/M-86 tries to protect the integrity of the open file. Try setting the applications program to [SYS,RO]. If this fails, refer to the Compatibility Attribute section of the <i>Concurrent CP/M-86 Operating System Programmer's Guide</i> and set attribute F3' on Set [F3 = on].</p> <p>This error also occurs when a process tries to open a file that is already open and the new open mode is incompatible with the mode in which the file was opened previously. Compatibility mode is determined by a parameter in GENCCPM, discussed in more detail in Section A.2.5 of this appendix.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
CP/M Error on d: File Opened in Read/Only Mode	<p>BDOS error. A process attempted to write to a file opened in Read-Only mode. A file is opened in Read-Only mode if it resides in User 0 and is opened from another user number. Locate the other process and either wait for it to finish, or abort it with a CTRL-C.</p> <p>For more information on modes of opening files, refer to Section A.2.5 of this appendix.</p>
CP/M Error on d: Illegal ? in FCB	<p>BDOS error. You used a wildcard character, a ? or an *, where wildcard filespecs are not permitted.</p>
CP/M Error on d: Invalid Drive	<p>BDOS error. The disk you selected in the command line is not valid in your system. Concurrent CP/M-86 supports 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.</p>
CP/M Error on d: No Room in System Lock List	<p>BDOS error. The Open File, Make File, and Access Drive function return this error when no room for new entries exists within the system lock list.</p> <p>The size of the system lock list is determined by a parameter in GENCCPM. Section A.2.5 of this appendix discusses GENCCPM in more detail.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
CP/M Error on d: Open File Limit Exceeded	<p>BDOS error. A program or process tried to open one more file than the maximum number of open files per process that the system can accommodate.</p> <p>The maximum number of files per process is set by GENCCPM. Refer to Section A.2.5 of this appendix for a more thorough discussion.</p>
CP/M Error on d: Password Error	<p>BDOS error. A password is incorrect or missing. Some utilities prompt you for the password; others do not. Avoid this error with password protected drives by setting your default password in this way: SET [default = password].</p>
CP/M Error on d: Read/Only File	<p>BDOS error. A process tried to write to a file when that file was set to Read-Only. Use the SET command, SET d:filespec [RW], to set the file to Read-Write, then reenter the command line.</p>
CP/M Error on d: Read/Only Disk	<p>BDOS error. A process tried to write to a file when the disk was set to Read-Only. Use the SET command, SET d:[RW], to set the disk to Read-Write, then reenter the command line.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
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 (for Yes), 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 (for No) 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 the section of this User's Guide on the PIP command for a discussion and list of the PIP options.
Directory needs to be reformatted for time/date stamps.	SET error. You attempted to set time/date stamps on an uninitialized directory. Refer to the INITDIR command in this User's Guide.
Disk I/O Error	ERA, ERAQ error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.
Drive Read Only	REN error. You attempted to rename a file on a drive that is set to read only. If you want to rename the file, you must first use the SET command to set the drive to Read-Write. Refer to SET [HELP] or the section in this User's Guide on the SET command for more information.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
<p>Enable password protection first: SET d:[protect=on]</p>	<p>SET error. You attempted to set a password or protection on a file without first enabling protection for the disk. Use the SET command, SET d:[protect=on], to enable protection, then reenter your SET command line. Refer to the section in this User's Guide on the SET command for a more thorough discussion of disk and file protection and passwords.</p>
<p>ERROR: BAD PARAMETER</p>	<p>PIP error. The command line contains an illegal parameter. Refer to "The PIP Command," for an explanation and listing of the PIP command parameters. Reenter the command with a valid parameter.</p>
<p>ERROR: CAN'T DELETE TEMP FILE - d:filespec</p>	<p>PIP error. This message indicates a damaged disk. Ensure that the disk is in place and retry the operation. If the error persists, the disk is damaged and must be replaced. Use PIP to copy as much of the damaged disk as possible to a fresh disk, and use your back-up disk, if you maintained one, to replace the lost files onto the new disk. If you do not have a back-up disk, you must recreate the lost files.</p>
<p>ERROR: CAN'T DELETE TEMP FILE INCOMPATIBLE MODE - d:filespec</p>	<p>PIP error. A temporary file, with the filetype .\$\$\$, exists and cannot be deleted because it is currently opened by another process. Check your consoles until you locate the process that has the file open and either wait for it to finish or abort the process with CTRL-C or the ABORT command from another console.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: CAN'T DELETE TEMP FILE INVALID PASSWORD - d:filespec	PIP error. You attempted to delete a password protected temporary file without providing the correct password. Type the correct password with the file specification, d:filename;password, and reenter the PIP command line.
ERROR: CAN'T DELETE TEMP FILE NONRECOVERABLE - d:filespec	PIP error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.
ERROR: CAN'T DELETE TEMP FILE R/O DISK - d:filespec	PIP error. A temporary file, with the filetype .\$\$\$, exists and cannot be deleted because the disk is set to Read-Only. Use the SET command, SET d:[RW], to set the disk to Read-Write, reenter the PIP command, and then use SET again, SET d:[RO], to reset the disk to Read-Only.
ERROR: CAN'T DELETE TEMP FILE R/O FILE - d:filespec	PIP error. A temporary file, with the filetype .\$\$\$, exists and cannot be deleted because it is set to Read-Only. Use the SET command, SET d:filespec [RW], to set the file to Read-Write, then reenter the PIP command.
ERROR: CLOSE FILE - d:filespec	PIP error. This message indicates a damaged disk. Ensure that the disk is in place and retry the operation. If the error persists, the disk is damaged and must be replaced. Use PIP to copy as much of the damaged disk as possible to a fresh disk, and use your back-up disk, if you maintained one, to copy the lost files onto the new disk. If you do not have a back-up disk, you must recreate the lost files.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: CLOSE FILE FCB CHECKSUM - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR: CLOSE FILE INVALID DISK SELECT - d:filespec	PIP error. The disk indicated in the file specification is not supported by your system configuration. Concurrent CP/M-86 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: CLOSE FILE NONRECOVERABLE - d:filespec	PIP error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.
ERROR: CLOSE FILE R/O DISK - d:filespec	PIP error. PIP cannot write the file indicated to the disk because the disk is set to Read-Only. Use the SET command, SET d:[RW], to set the disk to Read-Write, reenter the PIP command, and then use SET again, SET d:[RO], to reset the disk to Read-Only.
ERROR - Currently Opened	SET error. You attempted to use SET with a file that is currently open. Locate the process that has the file open and either wait for it to finish or abort it with a CTRL-C.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR - DIRECTORY FULL	<p>ED error. There is not enough directory space for the file being written. Switch to another console and use the ERA command to erase any unnecessary files on the disk. If the other consoles are busy and you do not want to interrupt the processes on any of them, you can use the "OXfilespec" command to erase unnecessary files without leaving the editor.</p> <p>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.</p> <p>Note: 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).</p>
ERROR - DISK FULL	<p>ED error. There is not enough disk space for the output file. This error might 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.</p>
ERROR: DISK READ - d:filespec	<p>PIP error. This message indicates a damaged disk. Ensure that the disk is in place and retry the operation. If the error persists, the disk is damaged and must be replaced. Use PIP to copy as much of the damaged disk as possible to a fresh disk, and use your back-up disk, if you maintained one, to copy the lost files onto the new disk. If you do not have a back-up disk, you must recreate the lost files.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: DISK READ CAN'T CLOSE CURRENT EXTENT - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR: DISK READ FCB CHECKSUM - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR: DISK READ INVALID DISK SELECT - d:filespec	PIP error. The disk indicated in the file specification is not supported by your system configuration. Concurrent CP/M-86 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: DISK READ INVALID FILENAME - d:filespec	PIP 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 the section of this User's Guide 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 command line.
ERROR: DISK READ NONRECOVERABLE - d:filespec	PIP error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: DISK READ RANDOM RECORD OUT OF RANGE - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR: DISK READ SEEK TO UNWRITTEN EXTENT - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR: DISK WRITE - d:filespec	PIP error. This message indicates a damaged disk. Ensure that the disk is in place and retry the operation. If the error persists, the disk is damaged and must be replaced. Use PIP to copy as much of the damaged disk as possible to a fresh disk, and use your back-up disk, if you maintained one, to copy the lost files onto the new disk. If you do not have a back-up disk, you must recreate the lost files.
ERROR: DISK WRITE CAN'T CLOSE CURRENT EXTENT - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR: DISK WRITE FCB CHECKSUM - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: DISK WRITE INVALID DISK SELECT - d:filespec	PIP error. The disk selected is not supported by your system configuration. Concurrent CP/M-86 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: DISK WRITE NO DATA BLOCK - d:filespec	PIP error. The disk to which PIP is writing is full. The file indicated by filespec was not copied. Reenter the PIP command line and redirect the output to a disk on another drive, if possible. If you do not have another drive available, use the ERA command to erase unnecessary files, if any, from the full disk or replace the disk before you reenter the PIP command line.
ERROR: DISK WRITE NO DIRECTORY SPACE - d:filespec	PIP error. The disk to which PIP is writing has no directory space available. Use the ERA command to erase unnecessary files, if any, from the full disk or replace the disk before you reenter the PIP command line. If you have another drive available, you could also redirect the output of the PIP command to the disk on the other drive.
ERROR: DISK WRITE NONRECOVERABLE - d:filespec	PIP error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: DISK WRITE R/O DISK - d:filespec	PIP error. The destination disk is set to Read-Only and PIP cannot write to it. Use the SET command, SET d:[RW], to set the disk to Read-Write, reenter the PIP command, and then use SET again, SET d:[RO], to reset the disk to Read-Only.
ERROR: DISK WRITE R/O FILE - d:filespec	PIP error. The destination file is set to Read-Only and PIP cannot write to it. Use the SET command, SET d:filespec [RW], to set the file to Read-Write, then reenter the PIP command.
ERROR: DISK WRITE RANDOM RECORD OUT OF RANGE - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR: DISK WRITE RECORD LOCKED - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR - File not available	ED error. The file you specified in the ED command line is currently opened by another process. Check your other consoles to locate the process that has the file open and either wait until it is finished or abort it with a CTRL-C or the ABORT command before you reenter the ED command line.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: FILE NOT FOUND - d:filespec	<p>PIP error. The input file indicated by 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.</p> <p>This error can also occur when the file you are trying to PIP has a system attribute. Use the R option to PIP this file, PIP d:filespec = d:filespec[R]. Refer to HELP PIP or to "The PIP Command" for more information.</p>
ERROR: FILE NOT FOUND INVALID DISK SELECT - d:filespec	<p>PIP error. The disk selected is not implemented in your system configuration. Concurrent CP/M-86 supports 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.</p>
ERROR: FILE NOT FOUND NONRECOVERABLE - d:filespec	<p>PIP error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.</p>
ERROR: First SET B:[protect=on]	<p>SET error. You tried to set or change a password before enabling password protection on the drive. SET B:[protect=on] will enable password protection on the drive and allow you to set and change passwords on the files on that disk. If the drive label itself has a password, you must provide it to change protection. Refer to "The SET Command" for a detailed explanation of password protection.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: HEX RECORD CHECKSUM - d:filespec	<p>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. Use the PIP command to copy the hex file from your back-up disks. If you do not have a back-up, you must erase the damaged file and recreate it. Refer to Section A.2.2 of this appendix for more information on recovery procedures for corrupted files.</p>
ERROR: Illegal command tail	<p>SDIR error. SDIR does not recognize what you entered after the command. Refer to HELP SDIR or to "The SDIR Command" for the correct syntax.</p>
ERROR: Illegal option or modifier	<p>SDIR error. The option or modifier you supplied is not valid with SDIR. Refer to HELP SDIR or to "The SDIR Command" for a listing and discussion of the SDIR options and modifiers.</p>
ERROR: Invalid Command Option	<p>SDIR error. The option or modifier you supplied is not valid with SDIR. Refer to HELP SDIR or the section in this User's Guide on the SDIR command for a listing and discussion of the SDIR options and modifiers.</p>
ERROR: INVALID DESTINATION - d:filespec	<p>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. Try the PIP command again, giving an output device or a specific file as the destination. Refer to the PIP command for a discussion of the valid PIP destinations.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: INVALID FORMAT	PIP error. The format of your PIP command is illegal. Refer to the PIP command for the correct syntax.
ERROR: INVALID FORMAT WITH SPARSE FILE - d:filespec	PIP error. This error usually occurs when you use PIP to concatenate or otherwise to change the format of a sparse file. Refer to the <i>Concurrent CP/M-86 System Programmer's Reference Guide</i> for more information on sparse files and their requirements.
ERROR: INVALID HEX DIGIT - d:filespec	PIP error. An invalid hex digit was encountered while reading a hex file. This message indicates a truncated or corrupted hex file. Use the PIP command to copy the file from your back-up disk. If you do not have a back-up, you must erase the damaged file and recreate it. Refer to Section A.2.2 of this appendix for more information on recovery procedures for corrupted files.
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. PIP interprets extra spaces as invalid characters.
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 "The PIP Command" for a discussion of the legal sources.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: INVALID USER NUMBER	PIP error. You have specified a user number greater than 15 or less than 0. Valid user numbers are in the range 0 to 15. Reenter the PIP command line using a valid user number.
ERROR: MAKE FILE - d:filespec	PIP error. This message indicates a damaged disk. Ensure that the disk is in place and retry the operation. If the error persists, the disk is damaged and must be replaced. Use PIP to copy as much of the damaged disk as possible to a fresh disk, and use your back-up disk, if you maintained one, to copy the lost files onto the new disk. If you do not have a back-up disk, you must recreate the lost files.
ERROR: MAKE FILE ALREADY EXISTS - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.
ERROR: MAKE FILE INVALID DISK SELECT - d:filespec	PIP error. The disk selected is not supported by your system configuration. Concurrent CP/M-86 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: Illegal Global/Local Drive Spec Mixing	SDIR error. This error occurs when you specify different drives in the filespec and the SDIR option. For example, SDIR A: [Drive=B] Omit one of the drive specifications

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: MAKE FILE INVALID FILENAME - d:filespec	PIP 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 2.5.1 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 command line.
ERROR: MAKE FILE LIMIT EXCEEDED - d:filespec	PIP error. There are too many files open for the Concurrent CP/M-86 open file list. Check your other consoles for processes with open files and either wait until they finish or abort a process with CTRL-C or the ABORT command.
ERROR: MAKE FILE NONRECOVERABLE - d:filespec	PIP error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.
ERROR: MAKE FILE R/O DISK - d:filespec	PIP error. PIP cannot create the file indicated because the disk is set to Read-Only. Use the SET command, SET d:[RW], to set the disk to Read-Write, reenter the PIP command, and then use SET again, SET d:[RO], to reset the disk to Read-Only.
ERROR: NO DIRECTORY SPACE - d:filespec	PIP error. The disk to which PIP is writing has no directory space available. Use the ERA command to erase unnecessary files, if any, from the full disk or replace the disk before you reenter the PIP command line. If you have another drive available, you could also redirect the output from the PIP command to the disk on the other drive.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: No HELP.HLP file on the default drive	HELP error. The HELP.HLP file must be on the default drive in order to access it.
ERROR: OPEN FILE - d:filespec	PIP error. This message indicates a damaged disk. Ensure that the disk is in place and retry the operation. If the error persists, the disk is damaged and must be replaced. Use PIP to copy as much of the damaged disk as possible to a fresh disk, and use your back-up disk, if you maintained one, to copy the lost files onto the new disk. If you do not have a back-up disk, you must recreate the lost files.
ERROR: OPEN FILE INCOMPATIBLE MODE - d:filespec	PIP error. Some other process has the file indicated by the variable, d:filespec, opened. Check your other consoles to locate the process and either wait until it has finished or abort it with a CTRL-C or the ABORT command before you reenter the PIP command. Section A.2.5 of this appendix discusses modes of opening files in more detail.
ERROR: OPEN FILE INVALID DISK SELECT - d:filespec	PIP error. The disk selected is not supported by your system configuration. Concurrent CP/M-86 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: OPEN FILE INVALID FILENAME - d:filespec	PIP error. This message indicates an error in the internal logic of PIP. Refer to Section A.2.6 of this appendix for instructions on reporting logic errors.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: OPEN FILE INVALID PASSWORD - d:filespec	PIP error. You tried to PIP a password protected file without entering the password. Include the password in the file specification and reenter the command line.
ERROR: OPEN FILE LIMIT EXCEEDED - d:filespec	PIP error. There are too many files open for the Concurrent CP/M-86 open file list. Check your other consoles for processes with open files and either wait until they finish or abort a process with CTRL-C or the ABORT command.
ERROR: OPEN FILE NONRECOVERABLE - d:filespec	PIP error. This error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.
ERROR: Parameter Required, try SET [HELP]	SET error. This message indicates a syntax error in the SET command line. Refer to the section of this User's Guide on the SET command for a discussion of the SET command syntax, or enter SET [HELP] for examples of correct SET syntax.
ERROR: PRINTER BUSY	PIP error. This message merely indicates that the printer is currently busy. Either wait until it is available, or locate the process using the printer on one of your other consoles and use a CTRL-C or the ABORT command to abort it.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: QUIT NOT FOUND	<p>PIP error. The string argument to a Q parameter was not found in your input file. Ensure that you typed the correct string and retry the operation. This error can also occur when entering the Q option from the command line. Concurrent CP/M translates everything in the command line into upper-case. 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.</p>
ERROR: RECORD TOO LONG	<p>PIP error. A hex record exceeds 80 characters in a file being copied with the [H] or [I] option.</p>
ERROR: Some other process has an open file.	<p>INITDIR error. INITDIR cannot be run when any program on any drive is accessing a file. Find the drive with the open file and either wait until the program is finished or abort it with a CTRL-C.</p>
ERROR: START NOT FOUND	<p>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 retry the operation. This error can also occur when entering the S option from the command line. Concurrent CP/M translates everything in the command line into upper-case. To avoid this problem, enter the S option from inside PIP. Type PIP, and wait for the * prompt before typing the file specifications and S option string.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: Too many entries in Index Table. Not enough memory	<p>HELP error. There is not enough memory available to hold the topic table while creating HELP.HLP. If you are running one or more large programs on another console, wait until they are finished before reentering the HELP [Create] command line. If HELP is the only program you are running, refer to Section A.2.4 of this appendix for more information on memory allocation.</p>
ERROR: Unable to close HELP.DAT Unable to close HELP.HLP	<p>HELP error. An error occurred while closing file HELP.HLP or HELP.DAT. The disk to which the file is being written is either full or has no directory space available. If the HELP.HLP file cannot be closed, go to another console and use the PIP command to copy the HELP.DAT file to another disk. Use the create option to create the HELP.HLP file on that disk, then erase HELP.DAT from the original disk. Use PIP to replace HELP.HLP on the original disk with the new HELP.HLP file.</p> <p>If the HELP.DAT file cannot be closed, use PIP to copy the HELP.HLP file to a fresh disk and set that disk to be the default drive. Enter the HELP [Extract] command and make your edits to the HELP.DAT file. Create the HELP.HLP file on the fresh disk, then use the PIP command to copy it to your system disk.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: Unable to Make HELP.DAT Unable to Make HELP.HLP	<p>HELP error. There is not enough space on the disk for HELP.HLP or HELP.DAT, or the files are Read-Only. If the files are Read-Only, use the SET command, SET d:filespec[rw], to set them to Read-Write. When you have finished, use SET again to set the new HELP.HLP file to Read-Only.</p> <p>If the HELP.HLP file cannot be made, go to another console and use the PIP command to copy the HELP.DAT file to another disk. Use the create option to create the HELP.HLP file on that disk, then erase HELP.DAT from the original disk. Use PIP to replace HELP.HLP on the original disk with the new HELP.HLP file. If the HELP.DAT file cannot be made, use PIP to copy the HELP.HLP file to a fresh disk and set that disk to be the default drive. Enter the HELP [Extract] command and make your edits to the HELP.DAT file. Create the HELP.HLP file on the fresh disk, then use the PIP command to copy it to your system disk.</p>
ERROR: UNEXPECTED END OF HEX FILE - {filespec}	<p>PIP error. The last hex record was not as long as the character count indicated. The hex file is truncated. Use the PIP command to copy the truncated file from your back-up disk. If you do not have a back-up, you must erase the damaged file and recreate it. Refer to Section A.2.2 of this appendix for more information on recovery procedures for corrupted files.</p>
ERROR: USER ABORTED	<p>PIP error. You aborted a PIP operation by entering a CTRL-C or by pressing a key. If you did not intend to abort the operation, you must reenter the PIP command line.</p>

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
ERROR: VERIFY - d: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. Usually, this error message indicates a hardware error. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.
ERROR: Writing file HELP.DAT	HELP error. An error occurred during extract while attempting to write the HELP.DAT file. This message usually means that the disk is full. Either erase any unnecessary files or copy the HELP.HLP file to a new disk and retry the operation.
ERROR: Writing file HELP.HLP	HELP error. An error occurred during create while attempting to write the HELP.HLP file. This message usually means that the disk is full. Either erase any unnecessary files or copy the HELP.HLP file to a new disk and retry the operation.
ERROR: Wrong Password	SET error. You tried to use SET with a file or drive that is password protected. The first time this message appears, a prompt for the password will appear under it. Type and enter the password. If it is incorrect, or if you do not enter it, this error message appears a second time and SET returns you to the system prompt.
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.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
File not found.	DIR, ED, ERA, ERAQ, TYPE error. The utility could not find the file indicated by the variable, filespec. Check the filename, drive code and user number before you reenter the command line.
File size out of range	VCMODE error. The buffer size you specified is too big. This message appears with examples and gives the legal range. Reenter the VCMODE command with a legal buffer size.
HELP.DAT not on current drive	HELP error. This error occurs when using the HELP [Create] option. The HELP.DAT file must be on the current drive. The HELP [Create] and HELP [Extract] commands do not work across drives. Change the current drive to the one with the HELP.DAT file and reenter the HELP [Create] command. The new HELP.HLP file will be created on that drive. If you want to replace your current HELP.HLP file with the new HELP.HLP, use the PIP command to copy the new file to your system disk.
Illegal State - system should be rebooted	VCMODE error. First try to reboot the system. If the error persists, it indicates an error in the internal logic of VCMODE. Refer to Section A.2.6 of this appendix for instructions on reporting internal logic errors.
Illegal time/date specification	DATE error. The time/date specification you entered has the wrong syntax. Refer to HELP DATE or the section in this User's Guide on the DATE command for the correct syntax and reenter the command line.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
Invalid Attribute	SET error. You tried to SET a drive attribute to a file, for example, time stamps, a label name, or protect=on. Refer to SET [HELP] or the section in this User's Guide on the SET command for a discussion and examples of the SET options.
Invalid Command Option	DIR, ERA, ERAQ, VCMODE error. The option or modifier you supplied is not valid. Refer to the HELP file or the section in this User's Guide on the command you are using for a listing and discussion of the command options and modifiers.
Invalid Filespec. Invalid Filespec (filespec)	ED, DIR, REN, TYPE error. The filename, filetype, password, or drive for the file you requested is incorrect. Correct the filespec and reenter the command line. Refer to the section in this User's Guide on naming files for more information on the correct forms of filenames and filetypes, and to the section on passwords for the correct form and usage of passwords.
Invalid Printer Number, IGNORED	TMP error. You specified an invalid printer number. The command was ignored. Specify a valid printer number and reenter the command line. Refer to the documentation provided by the manufacturer for the valid printer numbers for your particular system configuration.
Invalid User Number, IGNORED	TMP error. You have specified a user number greater than 15 or less than 0. User numbers are in the range 0 to 15. Reenter the command line using a valid user number.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
Invalid Wildcard	REN error. REN does not accept wildcards, ? or *, in file specifications, unless the wildcards are in exactly the same place in the old and new names, for example, REN newprog.* = oldprog.* works, but REN *.new = oldprog.* does not. Reenter the command line with a specific filename, or matching wildcards.
NO MEMORY	ED error. There is not enough space in available memory to create the buffer that ED requires to edit a file. If you are running one or more large programs on another console, wait until they are finished and reenter the command line. If there were no other programs running at the time this error occurred, you do not have enough memory in your particular hardware to run the program. Refer to Section A.2.4 of this appendix for more information.
No such file to rename	REN error. The file you specified as the oldfile in a REN command does not exist. You might have put the oldfile first and the newfile second. REN command syntax is: REN newfile = oldfile. Reenter the command line with the correct filename and the correct syntax.
No wildcards allowed.	TYPE error. The command specified in the command line does not accept wildcards in file specifications. Retype the command line using a specific filename.
Not erased: filespec Currently Opened	ERA, ERAQ error. Some other process is currently using the file you tried to erase. Either wait until the process is done or abort it before you reenter the ERA or ERAQ command line.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
<code>Not erased: filespec read only</code>	ERA, ERAQ error. You tried to erase a file that is Read-Only. Use the SET command to set the file to Read-Write. Refer to SET [HELP] or the section in this User's Guide on the SET command for more information.
<code>Not renamed: filespec already exists, delete (Y/N)</code>	REN error. You specified the name of an already existing file. If you enter a Y, the existing file will be deleted and the file you wish to rename will take its name. If you enter an N, nothing happens and Concurrent CP/M-86 returns the system prompt.
<code>Not renamed: filespec Currently opened</code>	REN error. The file you tried to rename is currently open on a different virtual console. Either wait for the program to finish or abort it before you reenter the REN command line.
<code>Not renamed: Invalid File</code>	This message indicates a command line syntax error. Refer to HELP REN or the section in this User's Guide on the REN command for the correct syntax and reenter the command line.
<code>Invalid Parameter</code>	SET error. The SET option you specified must have a filespec entered with it. Refer to SET [HELP] or the section in this User's Guide on the SET command for the correct syntax.
<code>Open files on drives d:</code>	BDOS error. This error occurs when you press a CTRL-C to abort a program if there are any open files on the drives mentioned. Because CTRL-C resets drives and aborts programs, this message is printed to warn you that you should not remove a disk until all files on that drive are closed.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
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.
Password Error	ERA, ERAQ, REN error. You specified a password-protected file without the password, or with an incorrect password. Reenter the command line with the correct password in the file specification, d:filespec;password.
Reading file HELP.HLP Reading HELP.HLP index	HELP error. An error occurred while reading HELP.HLP. This message indicates a damaged disk. Ensure that the disk is in place and retry the operation. If the error persists, the disk is damaged and must be replaced. Use PIP to copy as much of the damaged disk as possible to a fresh disk, and copy the HELP.HLP file from your back-up copy of the system disk.
REQUIRES CONCURRENT CP/M-86	DATE, DIR, ERA, ERAQ, HELP, PIP, SET error. This version of the utility can only be run under Concurrent CP/M-86 Version 2.0.

Table A-1. (continued)

<i>Message</i>	<i>Meaning</i>
Select Error	<p>SET error. This message indicates one of two errors:</p> <ul style="list-style-type: none"> ■ The drive specified is not supported by your system. Concurrent CP/M-86 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. ■ This error occurs when there is a hardware error on the disk. Refer to Section A.2.3 of this appendix for a list of possible hardware errors and their solutions.
Too many directory entries for query	<p>ERAQ error. You used a wildcard and specified too many files for ERAQ to query. Try breaking the group of files into two or more smaller groups, for example, all the files with a .DOC filetype and all the files with a .CMD filetype. Enter a separate ERAQ command for each group,</p> <pre>A>ERAQ *.CMD A>ERAQ *.DOC</pre>

A.2 Common Error Handling Procedures

This section discusses several classes of errors, their causes, and some recovery procedures. It is intended for the beginners who are interested in further details, more sophisticated users, and programmers needing more background information.

A.2.1 Checklist for Using Files

If you receive frequent error messages that Concurrent CP/M-86 cannot find your files, cannot find the commands you enter, or you cannot access files because they are password protected, you might use this checklist.

- Is the file set to DIR or SYS? If you do not understand file attributes, refer to "The SET Command."
 - ◆ Files with the DIR attribute can be accessed only if they are in the default user area on the default or specified drive except when using DIR, or PIP with the G option.
 - ◆ Files with the SYS attribute can be accessed if they are in the default user area or user 0 of the default or specified drive, or in the default or user 0 area of the system drive.
- Did you specify a drive when entering the command line? If a drive is specified, Concurrent CP/M-86 looks for a file only in the default and user 0 areas of the specified drive.
- Does the drive have a label? (Refer to "The SET Command" if you do not understand labels.) If the drive has a label and password protection is turned on for the drive, then you need a password to access any password protected files on that drive.
- Is the password protection mode set to READ, WRITE, DELETE, or NONE? (Refer to "The SET Command" if you do not understand the password protection mode.) Use SDIR to show the password protection already set for your files.

- ◆ If the password protection mode is set to READ, then you need a password to read the file.
- ◆ If the password protection mode is set to WRITE, you can read the file without supplying the password, but you need the password to write to the file.
- ◆ If the password protection mode is set to DELETE, you can read or write to the file, but you need the password to erase it.
- ◆ If the mode is set to NONE, the password is erased: you no longer need it at all.

The path Concurrent CP/M-86 uses to search for files is shown in the following table. The *Concurrent CP/M-86 Operating System Programmer's Reference Guide* discusses the file system in more detail.

Table A-2. Concurrent CP/M-86 Search Path

<i>Search Order</i>	<i>Drive Number</i>	<i>User Number</i>	<i>File Attribute</i>
1st	Default	Default	—
2nd	Default	0	SYS
3rd	System	Default	SYS
4th	System	0	SYS

A.2.2 Corrupted Files

Files can be corrupted by unforeseen events. Power outages are the most common cause. Replace the damaged file from your back-up copy. If you do not have a back-up copy, a command file can be replaced by reassembling the source code. Data files must be recreated by whatever method you used to create them the first time. If you are writing your own programs, it is possible to create one that writes over other files on the disk, or which corrupts the directory or other Concurrent CP/M-86 system tracks. If this class of error occurs after you have run a program of your own, debug your program carefully before attempting to run it again.

A.2.3 Hardware Errors Checklist

Hardware errors are typically disk errors. Either the disk is incorrectly inserted or the disk itself is damaged. Use the following checklist to locate the error.

- Is the disk correctly inserted?
 - ◆ If there is no disk in the drive, you might receive a BDOS Bad Sector, or Nonrecoverable error. Ensure that you have a disk in the drive before you continue.
 - ◆ If the disk is not securely in place or if the drive door is open you also receive BDOS Bad Sector or Nonrecoverable errors. Ensure that the disk is correctly inserted and the drive door securely closed before you continue.
 - ◆ This class of errors can also occur when the disk is not in the correct drive. If you are not quite sure which drive is the default drive and what letter designation it has (drive A, B, C, etc.), refer to the documentation provided with your particular system.
- Is the disk the correct density for the drive?
 - ◆ Some systems allow you to change the density of both drives and disk to single or double. Refer to the documentation provided by your particular system's manufacturer to find out if it is possible for you to change the density of your disks and drives. If your system allows you to change density, ensure that both the disk and the drive are set to the same density. Again, refer to the manufacturer's documentation for instruction on setting the density.
 - ◆ If you are using a single-sided disk in a drive set for double-sided disks, you will receive many strange errors. Check the manufacturer's documentation to ensure that your disks are correctly formatted for your particular system configuration.
- Is the disk physically write-protected? Your disk might be physically write protected. Usually, a tab can be punched out, or you can cover a notch with a piece of tape. For some reason, 5 1/4-inch diskettes are write-protected in the absence of a notch on the upper-right diskette edge, while 8-inch disks are write-protected by the presence of a notch on the bottom of the disk. Check the documentation provided by the disk manufacturer for instructions on removing the write-protection on the disk.

- If the error is not caused by one of the above problems, the disk might be damaged. Use the PIP command to copy as much of the information on the disk as possible to a fresh disk. Replace any files you cannot copy with the files from your back-up disk. If you have not maintained a back-up disk, you must recreate the missing files.
- If the error recurs, there might be a hardware problem with your system.
 - ◆ Check for loose connections.
 - ◆ If the connections all seem tight, the drive might be damaged. Contact the place you purchased your system for assistance.

A.2.4 No Memory

This error might also occur when there is space in memory, but the available memory is fragmented. Use the SYSTAT utility memory option to look at memory use. If you have enough total space available to run all the programs at the same time, try running them in a different order. If you simply do not have enough space, refer to the manufacturer's documentation or contact the place you purchased your system to see if it is possible to add more memory boards to your system configuration.

The *Concurrent CP/M-86 Operating System Programmer's Reference Guide* discusses memory allocation in more detail.

A.2.5 GENCCPM Options

A number of system characteristics, such as the PD table size, the system lock list, the open file limit, file opening modes, compatibility attributes, and several others, are determined by parameters specified when running GENCCPM.

GENCCPM is a system utility used by manufacturers to tailor Concurrent CP/M-86 to their particular system configuration. If you find you are receiving error messages regarding lack of space, files opened in incompatible modes, or if applications sold as running in the Concurrent CP/M-86 environment do not run under Concurrent CP/M-86, contact the customer support representatives at the place you purchased your system and at the manufacturer's offices (and, if necessary, consult the customer support representatives of the applications software's manufacturer).

Refer to the following section, "Reporting Logic Errors," for a list of the information you need to supply. The *Concurrent CP/M-86 Operating System System Guide* discusses GENCCPM in detail, and provides a systems programmer with the information needed to use GENCCPM to reconfigure your system.

A.2.6 Reporting Logic Errors

If you receive an error message indicating an error in the internal logic of the system or one of the utilities, contact the place you purchased your system for assistance. Be prepared to provide the following information:

- Indicate which version of the operating system you are using.
- Describe your system's hardware configuration: the number of drives, quantity of memory, and additional equipment like printers, modems, and so forth.
- Provide sufficient information to reproduce the error. Indicate which programs were running at the time the error occurred. If possible, provide a disk with a copy of the program.

End of Appendix A

Appendix B

Concurrent CP/M-86 User's Glossary

ambiguous filename: Filename that contains either of the Concurrent CP/M-86 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 Concurrent CP/M-86 file in a single command line.

applications program: Program that runs under an operating system. Typical applications programs are business accounting packages, word processing (editing) programs, mailing list programs, etc.

archive attribute: File attribute that indicates whether or not the file has been backed up. When you use PIP with the Archive option, it turns the archive attribute on. When a program changes a file, Concurrent CP/M-86 turns the archive attribute off, indicating that the file is new, and not backed up.

argument: A symbol, usually a letter, representing a number, letter, or name. Replace the symbol with an actual number, letter, or name to complete the command.

ASCII: ASCII refers to the American Standard Code for Information Interchange, a standard computer code for representation of numbers, letters, and symbols. An ASCII text file, such as the files produced by the ED editor, is a file that can be interpreted by your computer and intelligibly displayed on the video screen or printed on paper.

attribute: A file characteristic that can be set to on or off.

background/foreground: A virtual console is switched-in to the foreground, appearing on your screen, or switched-out to the background, with output invisible.

back-up: A copy of a disk or file made for safekeeping, or the creation of this disk or file.

bit: A switch in memory that can be set to on (1) or off (0). Eight bits grouped together comprise a byte.

block: An area in memory or space on a disk reserved for a specific use.

boot, bootstrap: The process of loading an operating system into memory and starting it up. 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 will manage. Typically, the boot is loaded automatically and executed at power up or when the computer is reset. Sometimes called a cold start. The name of the process comes from the concept of the computer starting itself "by its own bootstraps".

buffer: An area of memory or disk that temporarily stores data during the transfer of information.

Buffered mode: A switched-out mode where program output does not appear on your screen but is stored in a buffer. Contrast with **Physical mode** and **Dynamic mode**.

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

byte: A unit of memory or disk storage containing eight bits.

command: Elements of a Concurrent CP/M-86 command line. In general, a Concurrent CP/M-86 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 Concurrent CP/M-86 by typing the command keyword next to the system prompt on the console. The Concurrent CP/M-86 command files generally have a filetype of CMD. Files are either command files or data files. Same as a command program.

command keyword: A name that identifies a Concurrent CP/M-86 command, usually a file of type CMD, or the name of a Resident System Process. The command keyword precedes the command tail and the carriage return in the command line.

command syntax: A 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 you should replace with actual values when you enter the command.

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

concatenate: A term that describes one of PIP's operations: copying two or more separate files into one new file in a specified sequence.

console: The primary input/output device. The console is a screen and a keyboard through which you communicate with the operating system or applications program. Under Concurrent CP/M-86, a virtual console is one of four consoles sharing all system resources, including the screen and keyboard. See **virtual console**.

control character: A nonprinting character combination that sends a simple command to the currently executing process. To enter a control character, hold down the control (sometimes marked CTRL) key on your terminal and press the character key specified. In this document, the control key is represented by CTRL. A CTRL-X, for example, erases the command line. See Appendix C.

cursor: A 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: A nonexecutable collection of information used by the operating system or by an applications program.

default: The disk drive, user number, password, console, or list device obtained if you give any one of these without specifying a particular one. Any command that does not specify a disk drive or a user number references the default drive and default user number. When Concurrent CP/M-86 is first invoked, the default drive is the system drive. The default user number varies from console to console, until changed with the USER command. A default display is a display generated by a command keyword without any options.

delimiter: A special character that separates different items in a command line. For example, in Concurrent CP/M-86, a colon separates the drive specification from the filename. A period separates the filename from the filetype. A semicolon separates the filename and type from the password, and square brackets separate any options from their command or file specification. Commas separate one item in an option list from another. All of the above special characters are delimiters.

directory: A portion of a disk that contains entries for each file on the disk. In response to the DIR command, Concurrent CP/M-86 displays the filenames stored in the directory.

DIR attribute: A file attribute that causes a file to be accessible from the default user number and drive only.

directory label: Same as label. See label.

disk, diskette, floppy disk: Magnetic media for storing information. Programs and data are recorded on the disk in the same way that music is recorded on a cassette tape. The term diskette refers to the smaller capacity 5 1/4-inch removable floppy disk. Disk can refer to an 8-inch floppy disk, or a fixed (nonremovable) hard disk.

disk drive: A peripheral device that reads and writes on hard or floppy disks. Concurrent CP/M-86 assigns a letter to each drive under its control. For example, Concurrent CP/M-86 can refer to the drives in a four-drive system as A, B, C, and D.

drive label: Same as label. See label.

Dynamic mode: A switched-out operating mode where program output is not saved on a file. All but the last screenful of output is lost. Contrast with **Buffered mode** and **Physical mode**.

editor: A utility program that creates and modifies text files. An editor creates documents, data files, or source code for computer programs. The Concurrent CP/M-86 editor is invoked by typing the command ED next at the system prompt. (See the detailed ED description in Section 4 of this manual).

executable: An adjective meaning "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.

FCB: File Control Block. A structure defining file attributes, used by the system calls. Refer to the *Concurrent CP/M-86 Programmer's Reference Guide* for further details.

field: A portion of a record containing one data item such as a person's name, an address, or a phone number.

file: A collection of characters, instructions, or data stored on disk. You create document, data, or source files on disk with ED or a word processing applications program.

File Control Block: A structure defining file attributes, used by the system calls. Refer to the *Concurrent CP/M-86 Programmer's Reference Guide* for further details.

filename: The name assigned to a file. A filename is one to eight characters; a period separates the primary filename from the filetype. See the **file specification**.

file specification: A unique file identifier used by Concurrent CP/M-86 to locate a single specified file. A complete Concurrent CP/M-86 file specification includes a drive specification followed by a colon (d:), a filename of one to eight characters followed by a period, a filetype of zero to three characters followed by a semicolon, and a password. For example, b:example.tex;password is a complete Concurrent CP/M-86 file specification.

filetype: An extension to a filename, often defining the type of file. A filetype can be from zero to three characters and must be separated from the primary filename by a period. Certain programs require that files to be processed have certain filetypes.

floppy disk: A flexible magnetic disk used to store information. Floppy disks come in 5 1/4-and 8-inch diameters.

foreground/background: A virtual console is switched-in to the foreground, appearing on your screen, or switched-out to the background, with its output invisible.

global option: A use of parameters, delimited by square brackets, to affect more than one file in a directory or on a drive. Contrast with **local option**.

hard disk: A rigid, platter-like, magnetic disk sealed in a container. A hard disk stores more information than a floppy disk.

hardware: The physical components of a computer.

hex file: A hexadecimal 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 16 digits. Machine code is often converted to hexadecimal notation because it can be easily represented by ASCII characters and therefore printed on the console screen or on paper. In this manual we follow the industry standard of prefacing hexadecimal numbers with 0 and ending them with an "H": 0FFH is hexadecimal FF.

input: The data going into the computer, from the terminal or from a program reading from the disk.

interface: A device that allows two independent systems to communicate with each other, such as an interface between hardware and software in a microcomputer. The wire connecting two telephones could be considered an interface between them.

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.

label: An entry in the directory. The optional label contains information that describes special attributes of the disk to the operating system. For example, the label tells Concurrent CP/M-86 whether or not time stamping and password protection are turned on for that drive. You can give a label a name to help identify the data that is sorted on a given drive.

list device: A device, such as a printer, that lists or prints data.

local option: A use of parameters, delimited by square brackets, to affect a single file in a directory or on a drive. Contrast with **global option**.

logged in: Made known to the operating system, in reference to drives. A drive is logged in when it is selected by the user or an executing process and remains selected or logged in until a command is given to change the drive.

logical: A conceptual representation of something that might or might not be the same in its actual physical form. For example, a hard disk occupies one physical drive, yet you can divide the available storage on it to appear as if the data were in several different drives. Such apparent drives are called logical drives.

megabyte: Over one million bytes; 1024 kilobytes. See **byte**.

microprocessor: A semiconductor device that is the Central Processing Unit (CPU) of the microcomputer.

multiprogramming: The capability of the system to execute more than one program at a time. The microprocessor actually cannot execute multiple programs simultaneously, but it can switch between tasks so rapidly that it looks as if it is running programs concurrently.

multiuser: The ability of an operating system to support two or more users running programs at the same time.

operating system: The program that supervises the running of other programs and the management of computer resources. An operating system provides an orderly input/output environment between the computer and its peripheral devices.

option: A character or string appended to a command that modifies its operation. See **parameter**.

output: The data sent to the console, drive or printer.

parameter: A value in the command tail that provides additional information for the command. Technically, a parameter is a required element of a command. An option modifies the operation of a command; a parameter is data used by a command.

password: An extension to a filename that enables the user to add extra protection to his files by restricting access to it to those having the password. A password can be up to eight characters long and include any numeric, upper-, or lower-case characters and some special characters.

peripheral devices: The devices external to the processor itself. For example, terminals, printers, and drives are common peripheral devices that are not part of the processor, but are used in conjunction with it.

physical: The actual hardware of a computer. The physical environment varies from computer to computer.

Physical mode: A switched-in operating mode in which program output appears on your screen as the program generates it. Contrast with **Dynamic** and **Buffered** modes.

process: When Concurrent CP/M-86 runs programs, it refers to them as processes. See *The Concurrent CP/M-86 Programmer's Reference Guide* for further details.

program: A series of coded instructions that performs specific tasks when executed by a computer.

prompt: A character displayed on the video screen to help the user decide what the next appropriate action is. A system prompt is a special prompt displayed by the operating system; it indicates to the user that the operating system is ready to accept input. The Concurrent CP/M-86 system prompt is two characters followed by an angle bracket. The first character is numeric and indicates the default user number. The second character is alphanumeric and indicates the default drive. Some applications programs have their own special system prompts.

queue: Concurrent CP/M-86 uses data structures called queues as files in memory. Queues keep track of locked files, ready to run processes, and other system data, and also pass information between processes. Refer to the *Concurrent CP/M-86 Operating System Programmer's Guide* for further discussion.

Read-Only: Every file and drive is either Read-Only or Read-Write. When assigned to a file, the Read-Only attribute allows you to read from that file but not write any changes to it. When assigned to a drive, the Read-Only attribute allows you to read any file, but prevents you from adding a new file, erasing or changing a file, renaming a file, or writing on the disk. The SET command sets a file or a drive to Read-Only. The default setting for drives and files is Read-Write, but an error in setting disk density or other error found by Concurrent CP/M-86 automatically sets the drive to Read-Only until the error is corrected.

Read-Write: The Read-Write attribute allows you to read from and write to a specific Read-Write file or to any file on a drive set to Read-Write.

record: A collection of data. A Concurrent CP/M-86 record is 128 bytes long. A file consists of one or more records stored on a disk.

Resident System Process: A process that is made part of Concurrent CP/M-86 during system generation. An RSP can be associated with a queue accessed by a command keyword. The file containing the code might be stored as a file of type RSP.

R/O: Abbreviation for Read-Only.

RSP: Resident System Process.

run a program: Start a program executing. When a program is running, the computer is executing a sequence of instructions.

R/W: Abbreviation for Read-Write.

sector: A portion of a disk track. There are a specific number of sectors on each track.

software: All programs are software. They are machine readable instructions to the computer, as opposed to hardware, which is the actual physical components of a computer.

source file: An ASCII file that is an input file for a processing program, such as an editor, text formatter, assembler or compiler.

syntax: The format for entering a given command.

system attribute: An attribute assigned to a file enabling that file to be accessed from other than the default drive and user number. System files are generally placed on the system drive in user 0 because they are most easily and efficiently accessed from that location. A file can have either the SYS attribute or the DIR attribute.

system console: Same as console. A console that displays a Concurrent CP/M-86 system prompt.

system drive: The drive on which Concurrent CP/M-86 looks for files with the SYS attribute after checking the default or specified drive and failing to find them there.

systems program: Part of the operating system package, such as the Concurrent CP/M-86 utility programs described in this User's Guide.

system prompt: A symbol displayed by the operating system indicating that the system is ready to receive input. See prompt.

terminal: Generally, the same as a console.

time stamp: A record of when a file was created, accessed, or updated. In Concurrent CP/M-86, the SET command turns the time stamp on. The time and date information is appended to a file in the XFCB. The time stamps are displayed by the SDIR command.

TMP: Terminal Message Processor; a part of Concurrent CP/M-86 that manages the user interface, returns certain error messages, and so forth. See the *Concurrent CP/M-86 Operating System Programmer's Reference Guide* for further details.

track: The areas on a disk containing data, analogous to the grooves on a phonograph record. There are 77 tracks on a typical single-density eight-inch floppy disk.

turn-key application: Application designed for the noncomputer-oriented user. For example, a typical turn-key application is designed so that the operator needs only to turn on the computer, insert the proper program disk and select the desired procedure from a selection of functions (menu) displayed on the screen.

upward-compatible: Term meaning that a program created for the previously released operating system (or compiler, etc.) runs under the newly released version of the same operating system.

user number: Number assigned to a region of the directory so that different processes need only deal with their own files and have their own directories even though they are all working from the same drive. In Concurrent CP/M-86, there can be up to 16 users on a single drive.

utility: A program that enables you to perform certain operations, such as copying files, erasing files, and editing files. Utilities are created for the convenience of programmers and users. Concurrent CP/M-86 is distributed with over 25 utilities.

virtual console: The computer console that is switched-in, appearing on your screen, or switched-out and invisible. Programs run on individual virtual consoles regardless of whether they appear on the screen or not.

wildcard characters: The special characters that match certain specified items. In Concurrent CP/M-86 there are two wildcard characters, ? and *. The ? can be substituted for any single character in a filename, and the * can be substituted for the primary filename or the filetype or both. By placing wildcard characters in filenames, the user creates an ambiguous filename and can quickly reference one or more files.

XFCB: Extended File Control Block. XFCBs store passwords and time and date stamping of files. See FCB and Block in the *Concurrent CP/M-86 Programmer's Reference Guide*.

End of Appendix B

Appendix C

Concurrent CP/M-86 Control Character Summary

Table C-1. Concurrent CP/M-86 Control Characters

<i>Keystroke</i>	<i>Action</i>
CTRL-C	Prompts to abort a process currently running at a given console.
CTRL-E	Forces a physical carriage return, but does not send command to Concurrent CP/M-86.
CTRL-F	Flushes virtual console disk buffer.
CTRL-H	Same as BACKSPACE.
CTRL-J	Line-feed; terminates input at the console.
CTRL-M	Same as carriage return.
CTRL-O	Supresses current virtual console output until another CTRL-O is entered. CTRL-O is useful when you switch-in a background buffered console and want to avoid screen output.
CTRL-P	Echoes all console activity at the printer; a second CTRL-P ends printer echo. CTRL-P only works if your system is connected to a printer.
CTRL-Q	Resumes console display after CTRL-S.
CTRL-R	Retypes current command line; useful after using RUB or DEL key.
CTRL-S	Stops console display temporarily; CTRL-Q resumes the listing.

Table C-1. (continued)

<i>Keystroke</i>	<i>Action</i>
CTRL-U	Cancels line, displays #, cursor moves down one line and awaits a new command.
CTRL-X	Deletes all characters in command line.
CTRL-Z	String separator for PIP and ED; terminates console input when console is used as a source device with PIP.

End of Appendix C

Appendix D

Filetypes

Concurrent CP/M-86 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 specific kind of file. The following table lists common filetypes and their meanings.

Table D-1. Filetypes

<i>Filetype</i>	<i>Indication</i>
A86	Assembly language source file; the Concurrent CP/M-86 assembler, ASM-86, assembles or translates a file of type .A86 into machine language.
BAK	Back-up file created by a text editor; an editor renames the source file with this filetype to indicate that the original file has been processed. The original file stays on the disk as the back-up file, so you can refer to it.
CMD	Command file that contains instructions in machine executable code.
H86	Program file in hexadecimal format.

Table D-1. (continued)

<i>Filetype</i>	<i>Indication</i>
LST	Printable file that can be displayed on a console or printer.
PRN	Printable file that can be displayed on a console or printer.
SUB	Filetype required for SUBMIT program containing one or more Concurrent CP/M-86 commands. The SUBMIT program executes the commands in the submit file providing a batch mode for Concurrent CP/M-86.
TXT	Text file.
\$\$\$	Temporary file.

End of Appendix D

Appendix E

Drive and File Status Summary

The tables in this appendix tell you what information you receive with various combinations of the inquiry commands, DIR, SDIR, DIR[S], and SHOW. Each command, or combination of commands, provides you with quick information about the status of your files, drives or disks.

Table E-1. Display File Size

<i>Command</i>	<i># of Bytes in File</i>	<i># of Recs in File</i>	<i>Last Rec#</i>	<i>Free Diskette Space</i>	<i>Totals</i>
SDIR	X	X		X	X
SDIR [SIZE]	X				

Table E-2. Display File Attributes

<i>Command</i>	<i>SYS or DIR</i>	<i>RW or RO</i>	<i>SYS Files displayed?</i>	<i>User #</i>	<i>Archivel User Def. Attribute</i>
DIR [SYS]			X		
DIR filespec [SYS]			X		
SDIR	X	X	X	X	X

Table E-3. Display Time Stamping and Protection Modes

<i>Command</i>	<i>Creation or Last Access</i>	<i>Last Update</i>	<i>Password Mode</i>	<i>XFCB</i>
SDIR	X	X	X	X

Table E-4. Ways to Display Diskette, Label, and System Status

<i>Command</i>	<i>RO or RW</i>	<i>Free Diskette Space</i>	<i>Drive Display</i>	<i>Active Users</i>	<i>Active Files</i>	<i>Label Display</i>
SHOW	X	X				
SHOW d:	X	X				
SHOW SPACE	X	X				
SHOW DRIVE			X			
SHOW USERS				X	X	
SHOW LABEL						X

```
A>SHOW [DRIVE]
  A: Drive Characteristics
 1,248: 128 Byte Record Capacity
 156: Kilobyte Drive Capacity
 64: 32 Byte Directory Entries
 64: Checked Directory Entries
128: Records / Directory Entry
 8: Records / Block
 32: Sectors / Track
 1: Reserved Tracks

  B: Drive Characteristics
 1,248: 128 Byte Record Capacity
 156: Kilobyte Drive Capacity
 64: 32 Byte Directory Entries
 64: Checked Directory Entries
128: Records / Directory Entry
 8: Records / Block
 32: Sectors / Track
 1: Reserved Tracks
```

Figure E-1. Sample SHOW DRIVE Display

Directory Label	Passwds Req'd	Stamp Create	Stamp Update	Label Created	Label Updated
TOMSDISK.DAT	on	on	on	07/04/81 10:30	07/08/81 09:30

Figure E-2. Sample SHOW LABEL Display

End of Appendix E

151

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21



Appendix F

Differences Between Concurrent CP/M-86 2.0 and CP/M-86®

The primary difference between Concurrent CP/M-86 and CP/M-86 is that Concurrent CP/M-86 is a single-user, multitasking operating system, and CP/M-86 is a single-user, single-tasking system. Concurrent CP/M-86 provides all the mechanisms necessary to run more than one program at the same time and to prevent one task from interfering with another.

These mechanisms comprise most of the differences between Concurrent CP/M-86 and CP/M-86. Concurrent CP/M-86 allows two or more virtual consoles, while CP/M-86 allows one single console; Concurrent CP/M-86 does not allow file access to any file that is currently in use by any program—this means that if you are editing a file on one virtual console, you cannot edit the same file from another virtual console, nor can you type the file from another virtual console.

There are several differences between the utilities in CP/M-86 and Concurrent CP/M-86: Concurrent CP/M-86 includes time and date stamping (see DATE), and file password protection (see SET). ERAQ is a new utility, which erases a file as ERA does, but queries you before performing the erasure. HELP contains information to help you use the system, and also provides you with a place to maintain helpful hints of your own. VCMODE helps you manage some of the more complicated features of the virtual consoles. The functions STAT provided under CP/M-86 are replaced by various options of the SET, SDIR and SHOW utilities.

Unlike CP/M-86, Concurrent CP/M-86 uses a different memory allocation scheme, has queue management facilities, allows special file access (locked and unlocked files) and supports multitasking functions. Programming information is contained in two documents: *Concurrent CP/M-86 Programming Utilities Guide*, and *Concurrent CP/M-86 Programmer's Reference Guide*.

Concurrent CP/M-86 2.0 succeeds Concurrent CP/M-86 1.0. The major difference between the two versions is that Version 2.0 has a faster file system which means improved response time within programs. Time and Date stamping is implemented differently, thereby affecting the SET utility and adding the new INITDIR utility. The Version 1.0 utility, TOD, is renamed to DATE.

At the system level, the XIOS contains many differences. These are covered in the *Concurrent CP/M-86 Operating System System Guide*.

Note: the PIP utility shipped with Concurrent CP/M-86 version 1.0 is different from the version of PIP shipped with Concurrent CP/M-86 2.0. If you try to use the earlier version of PIP with Concurrent CP/M-86 2.0, it truncates files without warning.

Note: do not use the Concurrent CP/M-86 v.1.0 DSKMAINT utility with version 2.0 of Concurrent CP/M-86, because the XIOS calls are different.

Note: Concurrent CP/M-86 version 2.0 disks with time and date stamping enabled produce peculiar results if used under CP/M-86 1.0 or 1.1. You should be able to read these disks, but errors might result if you try to write to them. If you want to use a disk with both CP/M-86 and Concurrent CP/M-86, make sure time and date stamp space is recovered. Refer to the INITDIR utility for further details.

End of Appendix F

Index

!, 2-8
\$, 2-8, 3-54
\$\$, 3-54
\$\$\$ filetype, 3-23
[], 2-8
(), 2-8
, 2-8
., 2-8
SUB filetype, 3-54
/, 2-8
:, 2-8
;, 2-8
<, 2-8
=, 2-8
>, 2-8
?, 2-8

A

ABORT, 2-15, 3-7
access date, 3-53
ACCESS time stamp, 3-48
accessing files (see file access), 2-6
ambiguous filespec, 3-58
archive attribute, 3-39
archive option, 3-31
archives, 3-25
ASM-86, 2-6, 2-15
asterisk (*) as a wildcard, 2-8
attributes, 2-10
auxiliary device, 3-27

B

back-up copy, 2-2, 3-25, 3-31
background, 3-61
 (switched out), 1-3
basic editing commands, 4-8
booting, 2-1
 boot disk, 2-2
 boot loading, 2-1
buffered mode, 3-61

C

changing a filename, 3-36
character file, 3-22
character pointer, 4-8
character transfer, 3-26
colon, 2-3, 2-8
combining files, 3-26
command concepts, 2-1
 command line, 3-1
 command line notation, 3-4
 command specifications, 3-1
 command syntax, 3-3
commands, 3-54
 accessing files, 2-11
comment, 2-8
CON:, 3-27
concurrency, 1-2
console, 1-2
 virtual, 1-3
control character commands, 2-6
copy file, 3-22
copy files to and from auxiliary
 devices, 3-27
CREATE time stamp, 3-48, 3-53

creation date, 3-53
CTRL-C, 3-7, 3-55
CTRL-Z, 3-26
current user number, 2-10, 3-52

D

data files, 2-4
DATE, 2-15, 3-8
DDT-86, 2-15
default, 2-3
 drive, 2-3, 3-44
 format, 3-39
 label, 3-44
 password, 3-46
 user number, 3-59
delimiter, 2-7
DIR, 2-15, 3-10
 attribute, 2-10
directory, 1-4, 2-10, 2-12
 attribute, 3-39
 label, 3-39, 3-44, 3-45
 space in, 2-12
disk, 1-4
 buffer, 3-61
 damage, 2-2
 directory, 2-12
 drives, 1-2, 2-2 to 2-3
 sectors, 2-12
dollar sign symbol, 2-8, 3-55
drive, 1-4, 3-40
 attribute, 3-50
 designation, 2-3
 specification, 2-3 to 2-7, 3-1 to 3-2, 3-38, 3-51
dynamic allocation, 2-12
dynamic mode, 3-61

E

ED, 2-5, 2-15, 3-13
 error symbols, 4-31
 file error messages, 4-33
 prompt, 4-2
 text transfer commands, 4-5
editor, 4-1
end of file, 3-26
EOF:, 3-27
ERA, 2-15, 3-14
ERAQ, 2-15, 3-16
error messages, A-1
extended file control blocks, 3-41

F

file access, 2-4
file attributes, 1-5
 DIR, 2-11
 Read-Only, 2-11
 Read-Write, 2-11
 SYS, 2-11
 Archive, 2-11
file concatenation, 3-26
file creation, 2-1
file families, 2-5
file management, 1-2
file specification, 2-4, 2-6, 2-7, 3-1, 3-2
file structure, 2-1
filename, 2-4, 3-1, 3-2, 3-38
files, 1-2
 directory space, 2-12
 dynamic allocation, 2-12
 erasing, 2-12
 protecting, 2-11
filespec, 1-5, 3-3, 3-22
filetype, 1-5, 2-4, 2-8, 3-1 to 3-2, 3-38

floppy disk, 1-5
foreground (switched-in), 1-3
full format, 3-38, 3-39

G

GENCCPM, 2-15
GENCMD, 2-15
global options, 3-38
greater than, 2-8

H

hard disk, 1-5
HELP, 2-16, 3-17

I

I/O, 1-2
 management, 1-2
INITDIR, 2-16, 3-20, 3-48
insert mode, 4-15

K

keyboard, 2-3
kilobyte, 3-39 to 3-40, 3-51

L

labels,
 created, 3-53
 updated, 3-53
less than, 2-8
library file, 4-28

line editing control characters, 4-16
line numbers, 4-5, 4-22
logged in, 3-43
logical device, 3-26 to 3-27
lower-case, 3-30, 3-33, 3-34, 3-46
LST:, 3-27

M

machine object code, 3-26
memory buffer, 4-3
 size, 4-5
multiple command mode, 3-29
multiple file copy, 3-25
multitasking, 1-2

N

NUL:, 3-28

O

object file, 3-32
operating system, 1-1

P

parity bit, 3-33
passwords, 2-4, 2-8 to 2-9, 3-1, 3-16,
 3-39, 3-44, 3-52, 3-55
 password protection, 3-44 to 3-45,
 3-52
period, 2-8
peripheral device, 2-3
PIP, 2-6, 2-10, 2-16, 3-22, 3-29
power outage, 2-2
printer, 1-2

PRINTER, 2-16, 3-35

PRN:, 3-28

program files, 2-4

protecting files, 2-11

Q

question mark (?) as a wildcard, 2-8

R

R/O attribute, 2-11, 3-33

R/W attribute, 2-11

Read-Only, 3-33, 3-39 to 3-40, 3-44,
3-50 to 3-51

attribute, 2-11

Read-Write, 3-39, 3-50 to 3-51

attribute, 2-11

records, 3-39, 3-51

recovering from editing errors, 4-32

REN, 2-16, 3-36

rename, 3-36

repeated execution of editing
commands, 4-26

RO, 3-39

RW, 3-39

S

save editing changes, 4-6

SDIR, 2-8, 2-16, 3-38

searches, 3-38

sectors, 2-12

semicolon, 2-8

SET, 2-16, 3-1, 3-44

SET command, 2-10

SHOW, 2-16, 3-51

single file copy, 3-22

size format, 3-38, 3-39

slash symbol, 2-8

special characters in file
specifications, 2-7

square brackets, 2-8

stamp access, 3-53

stamp create, 3-53

stamp update, 3-53

storage space, 2-12

SUB file, 3-54

SUBMIT, 2-16, 3-54

SUBMIT parameters, 3-54

switched-in, 1-3

switched-out, 1-3, 3-61

syntax, 3-1

SYS, 3-22, 3-40, 3-44

attribute, 2-11

SYSTAT, 2-16

system

attribute, 3-39

drive, 2-4

files, 2-4, 3-32

prompt, 2-3 to 2-4

tracks, 3-25

T

temporary drive, 3-61

temporary file, 3-23

text editor, 2-6

text file, 3-58

time stamps, 3-39, 3-44, 3-48, 3-53

TYPE, 2-6, 2-16, 3-58

U

update time stamp, 3-53
upper-case, 3-30, 3-46
 translation, 4-15 to 4-16, 4-23
USER, 1-5, 2-16, 3-59
user-defined attributes, 3-39, 3-44
user number, 1-5, 2-4, 2-10 to 2-11,
 3-31, 3-40, 3-59
 changing, 2-10
 current user number, 2-10
utilities, 2-2

V

VCMODE, 2-16, 3-61
virtual consoles, 1-3, 1-4, 3-61

W

wildcard characters, 2-8, 3-3, 3-38,
 3-46
X\$\$\$\$\$\$\$.LIB file, 4-27
XFCB, 3-41

(

(

(

(