

**CP/M-68K™  
Operating System  
User's Guide**

**Copyright © 1983**

**Digital Research  
P.O. Box 579  
160 Central Avenue  
Pacific Grove, CA 93950  
(408) 649-3896  
TWX 910 360 5001**

**All Rights Reserved**

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.

#### 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 is a registered trademark of Digital Research. CP/M-68K, DDT, and DDT-68K are trademarks of Digital Research. EXORMacs is a registered trademark of Motorola Inc. IBM is a tradename of International Business Machines. Intel is a registered trademark of Intel Corporation. Motorola is a registered trademark of Motorola Inc.

The CP/M-68K Operating System User's Guide was prepared using the Digital Research TEX Text Formatter and printed in the United States of America.

\*\*\*\*\*  
\* First Edition: January 1983 \*  
\*\*\*\*\*

## Foreword

Welcome to the world of microcomputers opened to you by your Motorola® 68000 microprocessor. Welcome also to the world of application software accessible through your Digital Research CP/M-68K™ operating system. Digital Research designed CP/M-68K especially for the Motorola 68000 microprocessor that is the heart of your computer.

### **What CP/M-68K Does for You**

CP/M-68K manages and supervises your computer's resources, including memory and disk storage, the console (screen and keyboard), printer, and communications devices. It also manages information stored magnetically on disks by grouping this information into files of programs or data. CP/M-68K can copy files from a disk to your computer's memory, or to a peripheral device such as a printer. To do this, CP/M-68K places various programs in memory and executes them in response to commands you enter at your console.

Once in memory, a program runs through a set of steps that instruct your computer to perform a certain task. You can use CP/M-68K to create your own programs, or you can choose from the wide variety of CP/M-68K application programs that entertain you, educate you, and help you solve commercial and scientific problems.

### **What You Need to Run CP/M-68K on Your Computer**

The CP/M-68K operating system runs on Motorola's 68000 microprocessor. You need that microprocessor, a console device (generally a keyboard and display device such as a CRT screen), and at least one floppy disk drive. To use all the capabilities of CP/M-68K, you should have two disk drives. At least one should be a single density floppy drive compatible with the IBM 3740 diskette controller, because CP/M-68K and most CP/M applications are distributed on floppy disks suitable for this environment.

CP/M-68K and its utility programs are distributed on five floppy disks. The system disk, labeled 1 of 5, contains the operating system and the most commonly used utility programs. The second disk, labeled 2 of 5, contains additional utilities. Disks 3, 4, and 5 contain still more utilities, the C compiler, the C language library, files which can be used to build versions of CP/M-68K adapted to custom hardware environments, and sample BIOSes.

## How to Use CP/M-68K Documentation

The CP/M-68K documentation set includes four manuals:

- CP/M-68K Operating System User's Guide
- CP/M-68K Operating System Programmer's Guide
- CP/M-68K Operating System System Guide
- The C Programming Guide for CP/M-68K

The CP/M-68K Operating System User's Guide introduces you to the CP/M-68K operating system and tells you how to use it. The User's Guide assumes that the version of CP/M-68K on your distribution disk is ready to run on your computer. To use this manual, you must be familiar with the parts of your computer, you must know how to set it up and turn it on, and you must know how to handle, insert, and store disks. However, you do not need a great deal of experience with computers.

The CP/M-68K Operating System Programmer's Guide presents information for application programmers who are creating or adapting programs to run under CP/M-68K. The Programmer's Guide includes information on the CP/M-68K assembler and debugger that experienced programmers use to create new CP/M-68K programs.

The CP/M-68K Operating System System Guide describes the procedures required to adapt CP/M-68K for a custom hardware environment.

The C Programming Guide for CP/M-68K contains information for programmers who wish to use the C compiler and C language library included with CP/M-68K.

## How the User's Guide is Organized

This guide begins with simple examples, proceeds with basic concepts, then presents a detailed reference section on commands. The first four sections describe CP/M-68K operation for the first-time user.

Section 1 introduces CP/M-68K and tells you how to start the operating system, enter commands, edit the command line, and create a back-up copy of your distribution disks. Section 2 discusses files, disks, and drives. Section 3 describes how you can use CP/M-68K to manage your printer and console.

Section 4 develops the concepts you need to use CP/M-68K commands. If you are new to CP/M, read the first four sections carefully to get a general understanding of how to use CP/M-68K before you proceed to the specific command descriptions.

Section 5 provides detailed information on each CP/M-68K utility program, arranged alphabetically for easy reference. Section 6 tells you how to use ED, the CP/M-68K file editor. With ED, you can create and edit program, text, and data files.

Appendix A lists the messages CP/M-68K displays when it encounters special conditions, and describes corrective action where necessary. Appendix B provides an ASCII to hexadecimal conversion table. Appendix C lists the filetypes associated with CP/M-68K. Appendix D lists and defines the CP/M-68K control characters. Appendix E provides a simple glossary of commonly-used computer terms.

If you are new to computers, you might find some of the topics a bit difficult to understand at first. Learning to use your computer is a challenge, and we hope you will find it fun. This book proceeds step by step so that you can quickly proceed from opening your new system disk package to using CP/M-68K's powerful facilities.

# Table of Contents

|          |   |    |
|----------|---|----|
| <b>1</b> | <b>Introduction</b>                                       |    |
| 1.1      | How to Start CP/M-68K . . . . .                           | 1  |
| 1.2      | The Command Line . . . . .                                | 2  |
| 1.3      | Why You Should Back Up Your Files . . . . .               | 3  |
| 1.4      | How to Make Copies of Your CP/M-68K Disks . . . . .       | 4  |
| <br>     |   |    |
| <b>2</b> | <b>Files, Disks, and Drives</b>                           |    |
| 2.1      | What is a File . . . . .                                  | 7  |
| 2.2      | How are Files Created . . . . .                           | 7  |
| 2.3      | Naming Files--What's in a Name . . . . .                  | 8  |
| 2.4      | Accessing Files--Do You Have the Correct Drive? . . . . . | 9  |
| 2.5      | Accessing Files--User Numbers . . . . .                   | 10 |
| 2.6      | Accessing More Than One File . . . . .                    | 10 |
| 2.7      | File Attributes . . . . .                                 | 12 |
| 2.8      | How are Files Stored on a Disk? . . . . .                 | 12 |
| 2.9      | Changing Floppy Disks . . . . .                           | 12 |
| <br>     |   |    |
| <b>3</b> | <b>Console and Printer</b>                                |    |
| 3.1      | Controlling Console Output . . . . .                      | 15 |
| 3.2      | Controlling Printer Output . . . . .                      | 15 |
| 3.3      | Console Line Editing . . . . .                            | 15 |
| 3.4      | Assigning Logical Devices . . . . .                       | 17 |
| <br>     |   |    |
| <b>4</b> | <b>CP/M-68K Command Concepts</b>                          |    |
| 4.1      | Two Kinds of Commands . . . . .                           | 19 |
| 4.1.1    | Built-in Commands . . . . .                               | 19 |
| 4.1.2    | Transient Utility Commands . . . . .                      | 20 |

## Table of Contents (continued)

|          |  |    |
|----------|--|----|
| 4.2      | How CP/M-68K Searches for Program and Data Files . . .   | 22 |
| 4.2.1    | Finding Data Files . . . . .                             | 22 |
| 4.2.2    | Finding Program Files . . . . .                          | 23 |
| 4.3      | Executing Multiple Commands . . . . .                    | 24 |
| 4.4      | Terminating Programs . . . . .                           | 24 |
| 4.5      | Parts of a File Specification . . . . .                  | 25 |
| 4.6      | How Commands are Described . . . . .                     | 28 |
| <b>5</b> | <b>Command Summary</b>                                   |    |
| 5.1      | The COPY Command . . . . .                               | 33 |
| 5.2      | The Directory Commands: DIR and DIRS . . . . .           | 37 |
| 5.3      | The ED (Character File Editor) Command . . . . .         | 39 |
| 5.4      | The ERA Command . . . . .                                | 44 |
| 5.5      | The FORMAT Command . . . . .                             | 46 |
| 5.6      | The INIT Command . . . . .                               | 48 |
| 5.7      | PIP (Peripheral Interchange Program) Command . . . . .   | 49 |
| 5.7.1    | Single File Copy . . . . .                               | 49 |
| 5.7.2    | Multiple File Copy . . . . .                             | 51 |
| 5.7.3    | Combining Files . . . . .                                | 52 |
| 5.7.4    | Copy Files to and from Auxiliary Devices . . . . .       | 53 |
| 5.7.5    | Multiple Command Mode . . . . .                          | 55 |
| 5.7.6    | Using Options with PIP . . . . .                         | 56 |
| 5.8      | The REN Command . . . . .                                | 62 |
| 5.9      | The STAT (Status) Command . . . . .                      | 64 |
| 5.9.1    | Free Space on Disk . . . . .                             | 65 |
| 5.9.2    | Files - Display File Space and Access Mode . . . . .     | 65 |
| 5.9.3    | Set File Access Modes . . . . .                          | 68 |
| 5.9.4    | Display Disk Status . . . . .                            | 69 |
| 5.9.5    | Display User Numbers with Active Files . . . . .         | 70 |
| 5.9.6    | Display STAT Commands and Device Names . . . . .         | 70 |
| 5.9.7    | Display and Set Physical to Logical Device Ass . . . . . | 71 |

## Table of Contents (continued)

|          |   |     |
|----------|---|-----|
| 5.10     | The SUBMIT Command . . . . .                          | 72  |
| 5.10.1   | Creating the SUB File . . . . .                       | 72  |
| 5.10.2   | Executing the SUBMIT Command . . . . .                | 74  |
| 5.11     | The TYPE (Display File) Built-in . . . . .            | 77  |
| 5.12     | The USER ( Set User Number) Command . . . . .         | 78  |
| <b>6</b> | <b>The CP/M-68K Editor</b>                            |     |
| 6.1      | Introduction to ED . . . . .                          | 79  |
| 6.2      | Starting ED . . . . .                                 | 79  |
| 6.3      | ED Operation . . . . .                                | 80  |
| 6.3.1    | Appending Text into the Buffer . . . . .              | 82  |
| 6.3.2    | ED Exit . . . . .                                     | 83  |
| 6.4      | Basic Editing Commands . . . . .                      | 84  |
| 6.4.1    | Moving the Character Pointer . . . . .                | 85  |
| 6.4.2    | Displaying Memory Buffer Contents . . . . .           | 87  |
| 6.4.3    | Deleting Characters . . . . .                         | 88  |
| 6.4.4    | Inserting Characters into the Memory Buffer . . . . . | 89  |
| 6.4.5    | Replacing Characters . . . . .                        | 91  |
| 6.5      | Combining ED Commands . . . . .                       | 91  |
| 6.5.1    | Moving the Character Pointer . . . . .                | 92  |
| 6.5.2    | Displaying Text . . . . .                             | 92  |
| 6.5.3    | Editing . . . . .                                     | 93  |
| 6.6      | Advanced ED Commands . . . . .                        | 94  |
| 6.6.1    | Moving the CP and Displaying Text . . . . .           | 94  |
| 6.6.2    | Finding and Replacing Character Strings . . . . .     | 95  |
| 6.6.3    | Moving Text Blocks . . . . .                          | 98  |
| 6.6.4    | Saving or Abandoning Changes: ED Exit . . . . .       | 100 |
| 6.7      | ED Error Messages . . . . .                           | 101 |



# Appendixes

|   |   |     |
|---|---|-----|
| A | Error Messages . . . . .                    | 105 |
| B | ASCII and Hexidecimal Conversions . . . . . | 127 |
| C | File Types . . . . .                        | 131 |
| D | CPM-68K Control Character Summary . . . . . | 133 |
| E | User's Glossary . . . . .                   | 135 |

# Tables and Figures

## Tables

|      |  |     |
|------|--|-----|
| 3-1. | CP/M-68K Line-editing Control Characters . . . . . | 16  |
| 3-2. | CP/M-68K Logical Devices . . . . .                 | 17  |
| 4-1. | Built-in Commands . . . . .                        | 20  |
| 4-2. | Transient Utility Commands . . . . .               | 21  |
| 4-3. | Command Line Delimiters . . . . .                  | 27  |
| 4-4. | CP/M-68K Filetypes . . . . .                       | 27  |
| 4-5. | Command Line Conventions . . . . .                 | 29  |
| 5-1. | ED Command Summary . . . . .                       | 39  |
| 5-2. | PIP Options . . . . .                              | 57  |
| 6-1. | Text Transfer Commands . . . . .                   | 82  |
| 6-2. | Basic Editing Commands . . . . .                   | 85  |
| 6-3. | CP/M-68K Line Editing Controls . . . . .           | 90  |
| 6-4. | ED Error Symbols . . . . .                         | 102 |
| 6-5. | ED Disk File Error Messages . . . . .              | 103 |
| A-1. | CP/M-68K Error Messages . . . . .                  | 105 |
| B-1. | ASCII Symbols . . . . .                            | 127 |
| B-2. | ASCII Conversion Table . . . . .                   | 128 |
| C-1. | Common Filetypes . . . . .                         | 131 |
| D-1. | CP/M-68K Control Characters . . . . .              | 133 |

## Figures

|      |                                |    |
|------|--------------------------------|----|
| 6-1. | Overall ED Operation . . . . . | 81 |
|------|--------------------------------|----|

# Section 1

## Introduction to CP/M-68K

This section tells you how to start CP/M-68K, how to enter a command and edit the command line, and how to make a back-up copy of your CP/M-68K distribution disks.

### 1.1 How to Start CP/M-68K

Starting or loading CP/M-68K means reading a copy of the operating system from your CP/M-68K system disk (1 of 5 of your distribution disks) into your computer's memory. This is called a boot or cold start. Because the boot process varies from computer to computer, you must follow the manufacturer's instructions for the computer you use. Described in the next two paragraphs is a typical, but not universal, booting process.

First, be sure your computer's power is on. Next, insert the CP/M-68K system disk into your initial drive. In this section, assume that the initial drive is A and the disk is removable. Close the drive door. Then, press the RESET or RESTART button. This automatically loads CP/M-68K into memory.

If you want to restart CP/M-68K at some point after the initial loading of the system, first make sure your CP/M-68K system disk is in your initial drive then press the RESET or RESTART button. This is called system reset. It has the same effect as pressing RESET or RESTART when you first power up your computer. If you do a system reset after you have powered up, any data that is in your computer's memory prior to the system reset is erased. This means that any data that is not stored on a disk is lost.

After CP/M-68K is loaded into memory, the following message is displayed on your screen:

```
CP/M-68K          Version V.V  
Copyright(c) 1983 Digital Research Inc.
```

The version number, represented above by V.V, tells you the version of CP/M-68K that you own. After this display, the following two-character message appears on your screen:

```
A>
```

This is the CP/M-68K system prompt. The system prompt tells you that CP/M-68K is ready to read a command from your keyboard. In this example, the prompt also tells you that drive A is your default drive. This means that until you tell CP/M-68K to do otherwise, it looks for program and data files on the disk in drive A.

## 1.2 The Command Line

CP/M-68K performs certain tasks according to specific commands that you type at your keyboard. A CP/M-68K command line is composed of a command keyword, an optional command tail, and a RETURN keystroke. The command keyword identifies a command (program) to be executed. The command tail can contain extra information for the command, such as a filename or parameters. To end the command line, you must press the RETURN key. Note that the RETURN key can be marked ENTER, RETURN, CR, or something similar on your keyboard. In this guide, RETURN signifies the RETURN key.

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

You can type the keyword and command tail in any combination of upper-case and lower-case letters. CP/M-68K treats all letters in the command line as upper-case.

Generally, you type a command line directly after the system prompt. However, CP/M-68K does allow spaces between the prompt and the command keyword.

Let's use one command to demonstrate how CP/M-68K reads command lines. The DIR command tells CP/M-68K to give you a directory of the names of disk files on your screen, while the DIRS command displays only the system files. Type the DIRS keyword after the system prompt, omit the command tail, and press RETURN.

A>DIRS

CP/M-68K responds to this command by writing the names of those files that are stored in the default storage area. For discussion of storage areas, refer to Section 2.4. For example, if you have your CP/M-68K system disk in the default drive A, these filenames, among others, appear on your screen:

```
STAT      68K
PIP       68K
```

CP/M-68K takes the command keyword exactly as you type it and compares your input to executable program files in the CP/M-68K directory. If the command keyword matches a program file in the directory, CP/M-68K executes that program. If the first word you typed does not match any file in the directory, CP/M-68K echoes this unmatchable command keyword and puts a question mark after it. For example, if you mistype the DIRS command, CP/M-68K responds

A>DJRS  
DJRS?

to tell you that it can not find the command keyword. To correct simple typing errors, use the BACKSPACE key, or hold down the CTRL key and press H to move the cursor to the left. You cannot correct typing errors after you press the RETURN key. The command must then be typed correctly in its entirety. CP/M-68K supports other control characters that help you edit command lines efficiently. Section 3 tells how to use control characters to edit command lines and other information you enter at your console.

DIRS accepts a filename as a command tail. You can use DIRS with a filename to see if a specific file is on the disk. For example, to check that the transient utility program COPY.68K is on your system disk, type

A>DIRS COPY.68K

CP/M-68K performs this task by displaying either the name of the file you specified, or the message, No File.

Be sure to type at least one space after DIRS to separate the command keyword from the command tail. If you do not, CP/M-68K responds as shown.

A>DIRSCOPY.68K  
DIRSCOPY.68K? ..

### 1.3 Why You Should Back Up Your Files

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

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

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

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

So far, we have not discussed any commands that change information recorded on your CP/M-68K system disk. Before we do, let's make a few working copies of the your distribution disks.

#### 1.4 How to Make Copies of Your CP/M-68K Disks

To back up your CP/M-68K disks, you need several floppy disks. The back-up disks can be factory-fresh or used. You might want to format new disks or reformat used disks with the disk formatting program that accompanies your particular computer. If the disks are used, make sure that there are no files on the disk that you want to save.

If your computer's manufacturer has provided a special program to copy disks, you might use that program to make back-ups of your distribution disks. Otherwise, you can use the COPY utility program that comes with your set of CP/M-68K distribution disks. COPY is a track-by-track disk copying and formatting program that can copy the operating system loader and utility programs from your system disk. COPY creates a complete and exact copy of your system disk.

To use the COPY utility, simply type COPY and press the RETURN key. The following message comes on the screen:

```
COPY VER 1.1
```

followed by a mode menu:

| MODE  | FUNCTION                  |
|-------|---------------------------|
| ALL   | Copy the whole disk       |
| BOOT  | Copy the boot tracks only |
| FILES | Copy the files only       |
| END   | End the program           |

Once you have selected your mode, type it in and press RETURN. The following prompt appears on the screen:

```
Enter SOURCE drive: _
```

You must then type a drive letter from A to P and press RETURN. The program then prompts you to

```
Enter DESTINATION drive: _
```

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

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

The program gives you a chance to review what you have selected before proceeding. If you still want to proceed, press the RETURN key. If not, hold down the control key while pressing C and the process will be terminated. For an expanded treatment of the COPY utility, see Section 5, Command Summary.

End of Section 1

## Section 2

# Files, Disks, and Drives

CP/M-68K's most important task is to access and maintain files on your disks. With CP/M-68K you can create, read, write, copy, and erase disk files. This section tells you what a file is, how to create, name, and access a file, and how files are stored on your disks. It also tells how to change disks and change the default drive.

### 2.1 What is a File?

A CP/M-68K file is a collection of related information stored on a disk. Every file on a given disk must have a unique name because CP/M-68K uses that name to access that file. A directory is also stored on each disk. The directory contains a list of the files stored on that disk and the locations of each file on the disk.

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

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

### 2.2 How are Files Created?

There are many ways to create a file. You can create a file by copying an existing file to a new location, perhaps renaming it in the process. Under CP/M-68K, you can use the PIP command to copy and rename files. A second way to create a file is to use a text editor. The CP/M-68K text editor ED, described in Section 6, can create a file and assign it the name you specify. Finally, some programs such as AS68™ create output files as they process input files.



### 2.3 Naming Files--What's in a Name?

CP/M-68K identifies every file by its unique file specification. The simplest kind of file specification is a one- to eight-character filename, such as:

MYFILE

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

The drive specifier is a single letter (A-P) followed by a colon. Each drive in your system is assigned a letter. When you include a drive specifier as part of the file specification, you are telling CP/M-68K that the file is stored on the disk currently in that drive. For example, if you enter

B:MYFILE

CP/M-68K looks in drive B for the file MYFILE.

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

CUSTOMER

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

As you begin to use your computer with CP/M-68K, you will find that files fall naturally into categories. To help you identify files belonging to the same category, CP/M-68K allows you to add an optional one- to three-character extension, called a filetype, to the filename. When you add a filetype to the filename, separate the filetype from the filename with a period. Try to use three letters that tell something about the file's category. For example, you could add the following filetype to the file that contains a list of customer names:

CUSTOMER.NAM

When CP/M-68K displays file specifications in response to a directory or DIR command, it adds blanks to short filenames so that you can compare filetypes quickly.

The program files that CP/M-68K loads into memory from a disk have different filetypes, but are in the category of 68000 and equivalent programs that run with CP/M-68K. The filetypes .68K and .REL identify this category of executable programs.

We recommend that you create file specifications from letters and numbers. You must not use the following characters in filenames or filetypes because they have special meanings for CP/M-68K:

All Information Presented Here is Proprietary to Digital Research

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

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

A:DOCUMENT.LAW

## 2.4 Accessing Files--Do You Have the Correct Drive?

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

A>DIR COPY.68K

DIR looks in the directory of the disk in drive A for COPY.68K. If you have another drive, B for example, you need a way to tell CP/M-68K to access the disk in drive B instead. For this reason, CP/M-68K lets you precede a filename with a drive specifier. For example, in response to the command

A>DIR B:MYFILE.LIB

CP/M-68K looks for the file MYFILE.LIB in the directory of the disk in drive B. When you give a command to CP/M-68K, you should note which disk is in the default drive. Many application programs require that the data files they access be stored in the default drive.

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

A>B:PIP

CP/M-68K looks in the directory of the disk in drive B for the file PIP.68K. If CP/M-68K finds PIP on drive B, under user 0, it loads PIP into memory and executes it.

If you need to access many files on the same drive, you might find it convenient to change the default drive so that you do not need to enter a drive specifier repeatedly. To change the default drive, simply enter the drive specifier next to the system prompt. In response, CP/M-68K changes the system prompt to display the new default drive:

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

Because the drive specifier is part of a file's file specification, moving a disk to a different drive changes the file specifications of the files on that disk. Keep this in mind when moving disks from one drive to another.

Section 4 presents more information on how CP/M-68K locates program and data files.

## 2.5 Accessing Files--Do You have the Correct User Number?

CP/M-68K further identifies all files by assigning each file a user number, in the range from 0 to 15, when the file is created. The user number for each file is recorded in the disk directory. User numbers allow you to separate your files into sixteen file groups. User numbers are particularly useful for organizing hard disk space.

When you use a CP/M-68K utility to create a file, the file is assigned to the current user number, unless you use PIP to copy the file to another user number. You can determine the current user number by looking at the system prompt.

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

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

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

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

Most commands can access only those files that have the current user number. For example, if the current user number is 7, a DIR command displays only the files that were created under user number 7. However, if an executable program file or a SUBMIT file resides in user 0, the file can be accessed from any user number, in case it does not already exist in the current user area.

## 2.6 Accessing More Than One File

Certain CP/M-68K built-in and transient utilities can select and process several files when special wildcard characters are included in the filename or filetype. A file specification containing wildcards can refer to more than one file because it gives CP/M-68K a pattern to match. CP/M-68K searches the disk

directory and selects any file whose filename or filetype matches the pattern.

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

- A `?` matches any character in a name, including a space character.
- An `*` must be the last, or only, character in the filename or filetype. CP/M-68K internally replaces an `*` with `?` characters to the end of the filename or filetype.
- When the filename to match is shorter than eight characters, CP/M-68K treats the name as if it ends with spaces.
- When the filetype to match is shorter than three characters, CP/M-68K treats the filetype as if it ends with spaces.

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

A.68K, AA.68K, AAA.68K, B.68K, A.REL, and B.REL

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

|                             |   |
|-----------------------------|---|
| <code>*.*</code>            | is treated as <code>?????????.???</code>  |
| <code>?????????.???</code>  | matches all six names                     |
| <code>*.68K</code>          | is treated as <code>?????????.68K</code>  |
| <code>?????????.68K</code>  | matches the first four names              |
| <code>? .68K</code>         | matches A.68K and B.68K                   |
| <code>?.*</code>            | is treated as <code>? .???</code>         |
| <code>? .???</code>         | matches A.68K, B.68K, A.REL, and B.REL    |
| <code>A?.68K</code>         | matches A.68K and AA.68K                  |
| <code>A*.68K</code>         | is treated as <code>A?????????.68K</code> |
| <code>A?????????.68K</code> | matches A.68K, AA.68K, and AAA.68K        |

Remember that CP/M-68K uses wildcard patterns only while searching a disk directory, and therefore wildcards are valid only in filenames and filetypes. You cannot use a wildcard character in a drive specifier.

## 2.7 File Attributes

When you create a file, CP/M-68K gives it two attributes: DIR (for DIRectory) and R/W (for Read-Write). Through the STAT command, you can change these attributes from DIR to SYS (for SYStem) and from R/W to R/O (for Read-Only).

DIR and SYS attributes control whether CP/M-68K displays the file's name in response to a DIR command or DIRS command. DIR displays only the file specifications having the DIR attribute; DIRS displays only the file specifications having the SYS attribute.

The second file attribute can be set to either R/W (Read-Write) or R/O (Read-Only). If a file is marked R/O, any attempt to write data to that file produces a Read-Only error message. Therefore, you can use the R/O attribute to protect important files. A file with the R/W attribute can be read or written to at any time, unless you inadvertently switch disks during execution of a program.

## 2.8 How are Files Stored on a Disk?

CP/M-68K records the filename, filetype, user number, and attributes of each file in a special area of the disk called the directory. In the directory, CP/M-68K also records which parts of the disk belong to which file. You can use the STAT command to determine the number of entries in your directory.

CP/M-68K allocates directory and storage space for a file as records are added to the file. When you erase a file, CP/M-68K reclaims storage in two ways: it makes the file's directory space available to catalog a different file, and it frees the file's storage space for later use. This handling of storage and directory space, called dynamic file allocation, is a powerful feature of CP/M-68K. You do not have to tell CP/M-68K how big your file will become, because it automatically allocates more storage for a file as needed, and releases the storage for reallocation when the file is erased. Use the STAT command to determine how much space remains on the disk.

## 2.9 Changing Floppy Disks

CP/M-68K cannot, of course, do anything to a file unless the disk that holds the file is inserted into a drive and the drive is ready. When a disk is in a drive, it is on-line and CP/M-68K can access its directory and files.

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

You can also remove a disk and insert a new one when an application program prompts you to do so. This can occur, for example, when the data that the program uses does not fit on one floppy disk. Note that you must never remove a disk if a program is reading from or writing to it. See the CP/M-68K Programmer's Guide for more information on disk handling.

End of Section 2

## Section 3

# Console and Printer

This section describes how CP/M-68K communicates with your console and printer. It tells how to start and stop console and printer output, edit commands you enter at your console, and redirect console and printer input and output. It also explains the concept of logical devices under CP/M-68K.

### 3.1 Controlling Console Output

Sometimes CP/M-68K displays information on your screen too quickly for you to read it. Sometimes an especially long display scrolls off the top of your screen before you have a chance to study it. To ask the system to wait while you read the display, hold down the CTRL key and press S. A CTRL-S keystroke causes a pause in the display. When you are ready, press CTRL-Q to resume the display.

### 3.2 Controlling Printer Output

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

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

### 3.3 Console Line Editing

You can correct simple typing mistakes in the command line with the BACKSPACE or CTRL-H key. CP/M-68K also supports additional line-editing functions that you perform with control characters. You can use the control characters to edit command lines or input lines to most programs.

CP/M-68K allows you to edit your command line using the set of characters listed in Table 3-1. To edit a command line in CP/M-68K, use control characters to delete characters left of the cursor, then replace them with new characters.

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

A>POP A:=B:\*.\*)\_

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

A>P\_

Now you can type the correct letters, press RETURN, and send the command to CP/M-68K.

A>PIP A:=B:\*.\*)\_

Table 3-1 lists all line-editing control characters for CP/M-68K.

Table 3-1. CP/M-68K Line-editing Control Characters

| Character | Meaning   |
|-----------|---|
| CTRL-E    | Forces a physical RETURN but does not send the command line to CP/M-68K. Moves the cursor to the beginning of the next line without erasing your previous input.                |
| CTRL-H    | Deletes a character and moves the cursor left one character position. Has the same function as backspace.   |
| CTRL-I    | Moves the cursor to the next tab stop. Tab stops are automatically set at each eighth column. Has the same effect as pressing the TAB key.                                      |
| CTRL-J    | Sends the command line to CP/M-68K and returns the cursor to the left of the current line. Has the same effect as a RETURN or a CTRL-M.   |
| CTRL-M    | Sends the command line to CP/M-68K and returns the cursor to the left of the current line. Has the same effect as a RETURN or a CTRL-J.   |
| CTRL-R    | Places a # at the current cursor location, moves the cursor to the next line, and displays any partial command you typed so far. Pressing RETURN sends the command to CP/M-68K. |



Table 3-1. (continued)

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

You probably noticed that some control characters have the same meaning. For example, the CTRL-J and CTRL-M keystrokes have the same effect as pressing the RETURN key; all three send the command line to CP/M-68K for processing. Also, CTRL-H has the same effect as pressing the BACKSPACE key.

### 3.4 Assigning Logical Devices

Most CP/M-68K computer systems have a traditional console with a keyboard and screen display. Many also have letter-quality printers. If you use your computer for unusual tasks, you might want to add a different kind of character device to your system: a line printer, a teletype terminal, a modem, or even a joystick for playing games. To keep track of these physically different input and output devices, CP/M-68K associates different physical devices with logical devices. When you receive your CP/M-68K system, the logical devices are assigned to physical devices as shown in Table 3-2.

Table 3-2. CP/M-68K Logical Devices

| Logical Name | Device Type      | Physical Assignment                      |
|--------------|------------------|--|
| CON:         | Console input    | Keyboard                                 |
| CON:         | Console output   | Screen                                   |
| AUXI:        | Auxiliary input  | Auxiliary input port<br>(if one exists)  |
| AUXO:        | Auxiliary output | Auxiliary output port<br>(if one exists) |
| LST:         | List output      | Printer<br>(if one exists)               |

The assignment of logical names to physical devices is done by the hardware manufacturer. However, in some implementations of CP/M-68K, you can change these assignments with the STAT command. For example, you can assign AUXI and AUXO to a modem so that your computer can communicate with others over the telephone. The section on STAT in the Command Summary, Section 5, describes how to accomplish this.

End of Section 3

## Section 4

# CP/M-68K Command Concepts

As we discussed in Section 1, a CP/M-68K command line consists of a command keyword, an optional command tail, and a RETURN keystroke. This section describes the two kinds of programs the command keyword can identify, and tells how CP/M-68K searches for a program file on a disk. It also tells how to execute multiple CP/M-68K commands, and how to reset the disk system.

### 4.1 Two Kinds of Commands

A command keyword identifies a program that resides either in memory as part of CP/M-68K, or on a disk as a program file. Commands that identify programs in memory are called built-in commands. Commands that identify program files on a disk are called transient utility commands.

Seven built-in commands and several transient utility commands are included with CP/M-68K. You can add utilities to your system by purchasing various CP/M-68K-compatible application programs. If you are an experienced programmer, you can also write your own utilities that operate with CP/M-68K.

#### 4.1.1 Built-in Commands

Built-in commands are part of CP/M-68K and come into memory when you boot the system. Because they reside in memory, built-in commands have two advantages over transient commands:

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

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

Table 4-1. Built-in Commands

| Command | Function   |
|---------|--|
| DIR     | Displays filenames of files marked with the DIR (DIRectory) attribute in the directory.  |
| DIRS    | Displays filenames of files marked with the SYS (SYStem) attribute in the directory.   |
| ERA     | Erases a file from the disk by removing the filename from the directory and by releasing the storage space occupied by the file. |
| REN     | Renames a disk file.   |
| SUBMIT  | Executes a list of commands contained in a SUBMIT file that you create.  |
| TYPE    | Displays contents of an ASCII (character) file at your screen.   |
| USER    | Changes to a different user number.  |

CP/M-68K does not allow you to abbreviate the built-in commands.

#### 4.1.2 Transient Utility Commands

A transient utility command executes a program that comes into memory only when you request it. When you enter a command keyword that identifies a transient utility, CP/M-68K loads the program file from the disk and passes it any filenames, data, or parameters you entered in the command tail. Because these programs are loaded from disk, the disk with the program file must be accessible to execute the command. See Section 4.2 for an explanation of how CP/M-68K searches for command files.

Section 5 provides the operating details for the most frequently used CP/M-68K transient utilities listed in Table 4-2. Less frequently used utilities are described in the CP/M-68K Programmer's Guide. Many of these utilities are programming tools you might never need to use, however they are mentioned in Table 4-2.

Table 4-2. Transient Utility Commands

| Name   | Function   | Where Explained                        |
|--------|--|--|
| AR68   | Stores object files in the C run-time library: the Archive utility.  | CP/M-68K<br>Programmer's Guide         |
| AS68   | Invokes the assembler.   | CP/M-68K<br>Programmer's Guide         |
| C      | Invokes a submit file for calling the C language compiler.   | C Programming<br>Guide for 68K         |
| C068   | Intermediate compiling steps for the C language.   | C Programming<br>Guide for 68K         |
| C168   | Intermediate compiling steps for the C language.   | C Programming<br>Guide for 68K         |
| CP68   | Invokes the C Language Pre-processor for processing macros.  | C Programming<br>Guide for 68K         |
| COPY   | Copies disks (including CP/M-68K boot disks).  | CP/M-68K User's<br>Guide, Section 5    |
| DDT    | Invokes DDT™, the CP/M-68K debugger.   | CP/M-68K<br>Programmer's Guide         |
| DUMP   | Displays a file in ASCII and hexadecimal formats.  | CP/M-68K<br>Programmer's Guide         |
| ED     | Creates and alters character files.  | CP/M-68K User's<br>Guide, Section 5, 6 |
| FORMAT | Marks all disk sectors with the appropriate density and length.  | CP/M-68K User's<br>Guide, Section 5    |
| INIT   | Prepares a disk so that CP/M-68K can write on it. INIT erases any files and directory entries that are on the disk prior to execution. | CP/M-68K User's<br>Guide, Section 5    |
| LO68   | Invokes the Linker.  | CP/M-68K<br>Programmer's Guide         |
| NM68   | Invokes the NM68 utility that prints the symbol table.   | CP/M-68K<br>Programmer's Guide         |

Table 4-2. (continued)

| Name   | Function  | Where Explained                  |
|--------|---|----------------------------------|
| PIP    | Copies, combines, or transfers specified files between peripheral devices.  | CP/M-68K User's Guide, Section 5 |
| RELOC  | Relocates a command file containing relocation information to an absolute address.  | CP/M-68K Programmer's Guide      |
| SEND68 | Converts a command file to the Motorola S-record format.  | CP/M-68K Programmer's Guide      |
| SIZE68 | Prints the size of a command file.  | CP/M-68K Programmer's Guide      |
| STAT   | Shows the access status for disks or files, the amount of free space on disks, the space occupied by files, or the logical-to-physical assignment of devices, according to options specified in the command line. | CP/M-68K User's Guide, Section 5 |

## 4.2 How CP/M-68K Searches for Program and Data Files

This section describes how CP/M-68K searches for program and data files on disk. You need to understand the steps CP/M-68K follows, especially if it appears that CP/M-68K cannot find a program file you specified with a command keyword in a command line. If, for example, you have more than one disk drive on your system, the problem might be that CP/M-68K is not looking on the drive where the file is stored.

### 4.2.1 Finding Data Files

As you recall, when you enter a command line CP/M-68K passes the command tail to the program identified by the command keyword. If the command tail contains a file specification without a drive specification, the program asks CP/M-68K to search for the data file on the default drive and current user area.

For example, if you enter the following command line:

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

then CP/M-68K limits its search to user 3, drive A. If MYFILE.TXT has the DIR attribute and exists in user 3, drive A, CP/M-68K confirms the file's existence by displaying the filename on your screen:

**A:MYFILE.TXT**

Note that the user number does not appear in the directory listing. DIR also reports, System Files Exist, if user 3, drive A, contains files marked with the SYS attribute.

If MYFILE.TXT has the SYS attribute and exists in user 3, drive A, DIR merely reports that, System Files Exist; the filename MYFILE.TXT is not echoed to the screen.

Whether a data file is marked SYS or DIR, the search for a data file does not extend beyond the current user area and current drive. If CP/M-68K cannot find the data file, the program displays an error message at the console. Typically, this message is, File not found, or, No File; which message appears depends on the program identified by the command keyword.

**4.2.2 Finding Program Files**

When searching for a command file, CP/M-68K uses an extensive search pattern. This search is based on your command keyword and filetype which you may or may not supply.

If you add a filetype to your command keyword, CP/M-68K first searches the current user area for the command file with that filetype. If it cannot find the file in the current user area, CP/M-68K searches user area 0 for the command file having the filetype you supplied.

If you omit a filetype to your command keyword, CP/M-68K tries to supply one for you based on the following search pattern:

- CP/M-68K first checks the current user area and drive to see if there is a program file with your command keyword as its name and 68K as its filetype. If it does not, CP/M-68K then checks to see if it has a blank filetype. If the filetype is neither 68K nor blank, CP/M-68K then checks to see if the program file has a filetype of SUB.
- If the program file has none of these filetypes in the current user area, then CP/M-68K checks user area 0 for the same search pattern: first for the 68K filetype, then for the blank filetype, and finally for the SUB filetype.
- At any point in the search process, CP/M-68K stops the search if it finds the program file. CP/M-68K then loads the program into memory and executes it. When the program terminates, CP/M-68K displays the system prompt and waits for your next command.
- If CP/M-68K does not find the command file, it repeats the command, follows it with a question mark, and waits for your next command.

If you include a drive specifier before the command keyword, CP/M-68K first looks for the program file on the specified drive in the current user area. If the program file is not found there, CP/M-68K searches user area 0 on the same drive. The search goes no further than this when a drive specifier is included in the command keyword.

### 4.3 Executing Multiple Commands

In the examples used so far, CP/M-68K has executed only one command at a time. CP/M-68K can also execute a list of commands. You can enter a list of commands at the system prompt, or you can put a frequently needed list of commands in a disk file. Once you have stored the list in a disk file, you can execute the list whenever you need to with a SUBMIT command.

To list multiple commands on the command line, separate each command keyword and command tail from the next keyword with an exclamation point. When you complete the list, press RETURN. CP/M-68K executes your commands in sequence, as shown below:

```
3A>dirs!dir examp*.*!stat a:
```

CP/M-68K responds to the first of your three commands by reporting  
NON-SYSTEM FILE(S) EXIST

Without waiting for any operator intervention, CP/M-68K executes the second command:

```
3A>dir examp*.*
A: EXAMP7       : EXAMP1   TXT : EXAMP3       : EXAMP2   TXT : EXAMP4
A: EXAMP5       : EXAMP6
```

Finally it executes the third command in the series:

```
3A>stat a:
```

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

If you find you need to execute the same command list frequently, store the list in a disk file. See the description of the SUBMIT command in Section 5 for how to repeatedly execute a list of commands.

### 4.4 Terminating Programs

There might be times when you want to terminate a running program before it terminates itself. You can terminate the DIR, DIRS, TYPE, and SUBMIT built-in commands and all transient utilities, supplied with CP/M-68K whether invoked singly or in a multiple command line, simply by holding down the control key as



you press the C key. On many keyboards, the word control is abbreviated to CTRL. In addition, a CTRL-C can terminate many of the application programs that you run with your CP/M-68K operating system.

You enter a CTRL-C during program execution as the first character after the system prompt. CTRL-C does not work, for example, at the end of a command line, before you press RETURN.

If you try to abort a program other than DIR, DIRS, TYPE, or SUBMIT, you might experience a delay between the time you press CTRL-C and the time the program terminates because CP/M-68K checks CTRL-C only when reading from or writing to the console.

When you press CTRL-C during a program's execution CTRL-C causes a warm start, sometimes called a warm boot. Unlike other versions of CP/M, a warm start in CP/M-68K does not reload the operating system from your disk back into your computer's memory; CP/M-68K is always resident in memory. In CP/M-68K, a warm start does two things: it empties that portion of memory occupied by transient utilities and application programs, and it also resets all drives to a Read-Write state. At the end of a normal program execution of a transient utility or a built-in command, CP/M-68K performs a warm start.

#### 4.5 Parts of a File Specification

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

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

If you do not include a drive specifier, CP/M-68K automatically uses the default drive. If you omit the period and the filetype, CP/M-68K automatically includes a filetype of three blanks.

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

**B:MYFILE.DAT**

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

filespec

in the command syntax statements in Section 5.

Some CP/M-68K commands accept wildcards in the filename and filetype parts of the command tail. For example,

B:MY\*.A??

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

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

d:filename.typ

In the above form, d: represents the optional drive specifier, filename represents the one- to eight-character filename, and typ represents the optional one- to three-character filetype. The syntax descriptions in this section use the term filespec to indicate any valid combination of the elements included in the file specification. The following list shows valid combinations of the elements of a CP/M-68K file specification.

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

The characters in Table 4-3 have special meaning in CP/M-68K, so do not use these characters in a file specification except as indicated.

Table 4-3. Command Line Delimiters

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

CP/M-68K has already established several file groups. Table 4-4 lists some of their filetypes with a short description of each family. Appendix B provides the complete list.

Table 4-4. CP/M-68K Filetypes

| Filetype | Meaning                                      |
|----------|--|
| S        | assembler source file                        |
| 68K      | 68000 or equivalent machine language program |
| REL      | relocatable executable program file          |
| C        | C language source file                       |
| SUB      | list of commands to be executed by SUBMIT    |
| \$\$\$   | temporary file                               |

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

## 4.6 How Commands are Described

In Section 5, the Command Summary, CP/M-68K commands appear in alphabetical order. Each command description is given in a specific form. Section 5 also describes the notation that indicates the optional parts of a command line and other syntax notation.

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

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

- Words in capital letters must be spelled as shown, but you can use any combination of upper- or lower-case letters.
- A lower-case word in the syntax line has a general meaning that is defined in the text.
- The symbolic notation `d:`, `filename`, `.typ`, and `filespec` have the general meanings described in Section 4.5.
- You must include one or more space characters where a space is shown, unless otherwise specified. For example, the PIP options do not need to be separated by spaces.

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

Table 4-5. Command Line Conventions

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

Table 4-5. (continued)

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

Let's look at some examples of syntax notation. The CP/M-68K DIR (DIRectory) command displays the names of files catalogued in the disk directory.

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

```
Syntax:   DIR {d:} | {filespec}
           |         |
           optional optional
```

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

```
DIR
DIR john.ltr
DIR b:
DIR b:john.ltr
DIR b:john.*
DIR b:*.ltr
DIR b:*.*
```

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

```
DIR {d:} | {filespec}
```

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

The CP/M-68K command PIP (Peripheral Interchange Program) provides an example of a different kind of syntax. PIP can copy files from the disk to the screen or printer. PIP can combine two or more files into one longer file. PIP can also rename files after copying them. All of this functionality can be represented in the syntax

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

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

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

End of Section 4

## Section 5

# Command Summary

This section describes the commands and programs supplied with your CP/M-68K operating system. The commands are in alphabetical order. Each command is followed by a short explanation of its operation and examples. ED, the CP/M-68K editor, is described in detail in Section 6. Other commands, such as AS68 and DDT-68™, are described fully in the CP/M-68K Programmer's Guide.

### 5.1 The COPY (copy disk) Command

Syntax: COPY {option list}

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

Unlike PIP, which needs to know both the type and length of file to be copied, COPY is not sensitive to the types of files or their lengths. Even so, COPY requires that your destination disk have the same format as your source disk. Hence you may have to format a destination disk before using COPY. If so, see FORMAT for details.

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

Copy Ver 1.1

A mode menu then appears:

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

Choose one mode from the mode menu. For example, if you want to copy everything on one of your CP/M-68K disks, choose ALL to copy CP/M-68K and all files



beyond the boot tracks. Choose BOOT if you want to copy only the boot loader for CP/M-68K. Choose FILES if you want to copy only the non-boot tracks. Choose END if you wish to leave the COPY program and return to the command level of CP/M-68K.

Once you have selected your mode, COPY prompts you with

Enter SOURCE drive:    

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

Similarly, you get the prompt:

Enter DESTINATION drive:    

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

(^C to ABORT)

RETURN to copy <mode> from <source> to <destination>.

Mode, source, and destination have actual values on your screen in place of the words shown here. If these are the values you truly want COPY to use, press the RETURN key. COPY begins copying at this point, as indicated by the message, \*\*\*Copying nn Tracks\*\*\*, that appears at the bottom of your screen. The nn in this prompt stands for the actual track numbers currently being copied.

If you decide to abort the copy process, type CTRL-C (that is, hold down the CONTROL key and press the letter C). If you have not used the [A] (for automatic) option when you were at the CP/M-68K command line, then you are offered a chance to do some more copying when COPY finishes your first copy. COPY saves the parameters you gave it for the last copy just made; then COPY asks you:

Do you wish to repeat the copy?    

Type y if you do, n if you do not. If you type n, the COPY program ends without any further processing and returns you to the CP/M-68K command level. If you type y, COPY shows you the values used in the last copy session and invites you to start another copy session by entering a carriage return. For example, if you chose mode ALL, source drive A, and destination drive B in your initial copy session, COPY returns the following prompt for all subsequent copy sessions:

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

If you decide at this point not to begin a new session, type CTRL-C. CTRL-C returns you to the command level of CP/M-68K.

Examples:

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

```
{mode option} before
{source drive option} before
{destination drive option} before
{automatic carriage return option}
and/or {verify option} before
{manual carriage return}
```

Some representative combinations of the COPY command line appear below:

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

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

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

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

With the exception of the A and V options, which are discussed below, each of the options in the above examples is explained in the preceding basic, interactive COPY session.

**Note:** Be sure the source drive appears to the left of the destination drive in your option list. If you inadvertently get the the drive letters of your destination drive and source drive turned around in this option list, you will copy your destination disk (probably blank) on to your source disk, thereby wiping it out. Avoid this by entering the source drive letter before you specify the destination drive letter. Then recheck what you typed before you press the RETURN key.

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

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

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

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

the COPY utility loads then displays the message:

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

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

## 5.2 The DIR (Directory) and DIRS (System Directory) Commands

Syntax:

```
DIR {d:}
DIR {filespec}

DIRS {d:}
DIRS {filespec}
```

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

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

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

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

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

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

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

No File

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

SYSTEM FILE(S) EXIST

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

NON-SYSTEM FILES(S) EXIST

Examples:

A>DIR

Displays all DIR files catalogued in user 0 on the default drive A.

A>DIR B:

Displays all DIR files for user 0 on drive B.

A>DIR B:X.BAS

Displays the name X.BAS if the file X.BAS is present on drive B.

4A>DIR \*.BAS

Displays all DIR files with filetype BAS for user 4 on drive A.

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

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

A>DIRS

Displays all files for user 0 on drive A that have the system (SYS) attribute.

3A>DIRS \*.68K

This wildcard form of the DIRS command displays all SYS files with filetype 68K on the default drive A for user 3.

### 5.3 The ED (Character File Editor) Command

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

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

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

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

The ED utility uses a portion of your user memory as the active text buffer where you add, delete, or alter the characters in the file. You use the A command to read all or a portion of the file into the buffer. You use the W or E command to write all or a portion of the characters from the buffer back to the file.

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

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

**Table 5-1. ED Command Summary**

| Command | Action  |
|---------|---|
| nA      | Append n lines from original file to memory buffer.                         |
| 0A      | Append file until buffer is one half full or until end of input is reached. |

Table 5-1. (continued)

| Command     | Action   |
|-------------|--|
| #A          | Append file until buffer is full (or end of file).                       |
| B, -B       | Move CP to beginning (B) or bottom (-B) of buffer.                       |
| nC, -nC     | Move CP n characters forward (C) or back (-C) through buffer.            |
| nD, -nD     | Delete n characters before (-D) or from (D) the CP.                      |
| E           | Save new file and return to CP/M-68K.                                    |
| Fstring{↑Z} | Find character string.   |
| H           | Save the new file, then reedit, using the new file as the original file. |
| I           | Enter insert mode; use ↑Z to exit insert mode.                           |
| Istring{↑Z} | Insert string at CP.   |

Table 5-1. (continued)

| Command   | Action  |
|---|---|
| <code>Jsearch_str^Zins_str^Zdel_to_str{↑Z}</code> | Juxtapose strings.  |
| <code>nK, -nK</code>                              | Delete (kill) n lines from the CP.                        |
| <code>nL, -nL, OL</code>                          | Move CP n lines.  |
| <code>nMcommands</code>                           | Execute commands n times.                                 |
| <code>n, -n</code>                                | Move CP n lines and display that line.                    |
| <code>n:</code>                                   | Move to line n.   |
| <code>:ncommand</code>                            | Execute command through line n.                           |
| <code>Nstring{↑Z}</code>                          | Extended find string.                                     |
| <code>O</code>                                    | Return to original file.                                  |
| <code>nP, -nP</code>                              | Move CP 23 lines forward and display 23 lines at console. |



Table 5-1. (continued)

| Command                           | Action  |
|-----------------------------------|---|
| Q                                 | Abandon new file, return to CP/M-68K.             |
| R                                 | Read X\$\$\$\$\$\$\$.LIB file into buffer.        |
| Rfilespec{↑z}                     | Read filespec into buffer.                        |
| Sdelete string^zinsert string{↑z} | Substitute string.                                |
| nT, -nT, 0T                       | Type n lines.                                     |
| U, -U                             | Upper-case translation.                           |
| V, -V, 0V                         | Line numbering on/off, display free buffer space. |
| nW                                | Write n lines to new file.                        |
| nX                                | Write or append n lines to X\$\$\$\$\$\$\$.LIB.   |

Table 5-1. (continued)

| Command                     | Action   |
|-----------------------------|--|
| <code>nXfilespec{↑Z}</code> | Write <code>n</code> lines to <code>filespec</code> or append if previous <code>x</code> command applied to the same file. |
| <code>0X</code>             | Delete file <code>X\$\$\$\$\$\$\$.LIB</code> .   |
| <code>0Xfilespec{↑Z}</code> | Delete <code>filespec</code> .   |
| <code>nZ</code>             | Wait <code>n</code> seconds.   |

## 5.4 The ERA Command

Syntax: ERA {filespec}

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

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

Command lines that take the form:

ERA {d:}wildcard-filespec

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

Confirm (Y/N)?

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

There is one instance in which the ERA command with a wildcard does not cause the system to prompt you. In a SUBMIT file, when you use a wildcard with an ERA command, CP/M-68K does not ask for confirmation. Please see Section 5.10 for a discussion of the SUBMIT command.

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

No File

Examples:

A>ERA X.PAS

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

A>ERA \*.PRN

The system asks to confirm:

Confirm (Y/N)?Y

All files with the filetype PRN are removed from the disk in drive A, user 0.

```
A>ERA B:*. *  
Confirm (Y/N)?Y
```

All files on drive B, user 0, are removed from the disk.

## 5.5 The FORMAT Command

**Syntax:**       FORMAT {disk drive}

**Explanation:** Digital Research provides a format utility, called FORMAT, that is compatible with CP/M-68K and the Motorola EXORMacs computer. The EXORMacs hardware is described in the CP/M-68K Operating System System Guide.

Most CP/M-68K users do not have an EXORMacs. Consequently, their disk formatting programs are supplied by the vendors of their non-EXORMacs hardware. Just the same, the formatting concepts mentioned above apply to all disk users, whatever brand of hardware is used.

Formatting establishes the desired disk density (either single density or double density), the desired sector size (128, 256, 512, or 1024), and the number of tracks appropriate for the disk to be formatted. Formatting makes a disk ready to receive new data from another disk having the identical format. Thus formatting is a preliminary, and often necessary, step to copying.

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

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

Executing FORMAT at the CP/M-68K command level takes the form:

```
A>FORMAT d
```

where d specifies a disk in the range a...p.

As with INIT, FORMAT asks you to reconfirm your choice in the following prompt:

Do you really want to format disk d?

Press y for yes, in either upper-case or lower-case, if you wish to continue. Reply with n for no, if you decide not to format disk d. If you reply to the above question with any character other than y, FORMAT aborts execution and returns you to the command level of CP/M-68K.

## 5.6 The INIT Command

Syntax:           INIT {disk drive}

Explanation:   INIT initializes disk directories. That is, INIT purges the contents of an existing directory by marking each sector in the directory area as unused, ready to be written on. Once this happens, data files referenced by the initialized directory are virtually erased. The effect is very similar to ERA \*.\* , but more thorough than ERAse because INIT can access every directory entry, no matter how bizarre the directory entry may be. Therefore, the primary use of INIT is to clean up a corrupted directory. If this is your purpose, be sure to PIP any important files from the disk to be initialized on to a back-up disk, then use INIT.

Note that INIT cannot change the size of your sectors or the density of your disk. In other words, INIT cannot reformat anything. INIT is hardware independent; it can be used on any computer equipped with Motorola's 68000 CPU and Digital Research's CP/M-68K operating system.

Executing INIT from the CP/M-68K command level takes the form:

**A>INIT d**

where d specifies a drive in the range a...P.

If initialization erases the contents of all directory entries on disk, CP/M-68K seeks reconfirmation of your intent before it proceeds:

Do you really want to initialize disk d?

Reply y for yes, or n for no, to the question. INIT proceeds to initialize your disk if you reply y. Reply with n for no if you decide not to initialize disk d. If you reply to the above question with any character other than y, INIT aborts execution and returns you to the command level of CP/M-68K.

## 5.7 PIP (Peripheral Interchange Program - Copy File) Command

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

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

### 5.7.1 Single File Copy

Syntax:

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

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



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

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

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

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

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

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

You can include PIP options following each source name (see PIP Options, below). There is one valid option, [Gn] - go to user number n, for the destination file specification. Options are enclosed in square brackets. Several options can be included for the source files. They can be packed together or separated by spaces. Options can verify that a file was copied correctly, allow PIP to read a file with the 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 much more.

Examples:

```
A>PIP B:=A: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. When using this form you cannot copy a file from one drive and user number to the same drive and user number. You must put the destination file on a different drive or in a different user number. See the section on PIP Options, and the section on the USER command. The second short form produces exactly the same result as the first one. PIP simply looks for the file oldfile.dat on drive A, the drive specified as the source.

**A>PIP B: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 names a file 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 (usually user 0 because CP/M-68K automatically starts out in user 0 - the default user number) to the same drive and user number. This effectively gives you two copies of the same file on one drive and user number, each with a different name.

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

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

**B>PIP program2.dat = A:program1.dat[E V G3]**

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

### 5.7.2 Multiple File Copy

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

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

Examples:

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:\*.\*

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 CP/M-68K bootstrap loader from the system tracks. COPY, or any other track-to-track copy program, copies the system tracks for you.

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

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

A>PIP B:[G1]=A:\*.A86

This command causes PIP to copy all the files with a filetype of A86 on drive A in the default user number, user ZERO in this case, to drive B in user number 1. Remember that the DIR, TYPE, ERA and other commands only access files in the same user number from which they were invoked. See the USER command.

## 5.7.3 Combining Files

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

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

it in the user number specified by n. You can specify one or more options for each source file.

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

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

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

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

When concatenating files, PIP only searches 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 code files. The [O] option causes PIP to ignore embedded CTRL-Z (end-of-file) characters, normally used to indicate the end-of-file character in 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 upon successful completion of the copy operation. 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 when PIP runs to successful completion.

#### 5.7.4 Copy Files to and from Auxiliary Devices

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

|      |                  |
|------|------------------|
| AXO: | AXI: {[options]} |
| CON: | CON: {[options]} |
| PRN: | NUL:             |
| LST: | EOF:             |

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

The logical device names are listed as follows:

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

AXI: Auxiliary Input or Output Device.

AXO: Auxiliary Output Device.

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

There are three device names that have special meaning:

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

EOF: A virtual source device that produces a single CTRL-Z, the CP/M-68K end-of-file mark.

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

Examples:

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

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

A>PIP B:FUNFILE.SUE=CON:

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

A>PIP LST:=CON:

If CRT: is assigned to CON:, 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[T8]

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 sixty lines.

### 5.7.5 Multiple Command Mode

Syntax: PIP

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

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

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

Terminate PIP by pushing only the RETURN key following the asterisk prompt. The empty command line tells PIP to discontinue operation and return to the CP/M-68K system prompt.

Examples:

```
A>PIP
*NEWFILE=FILE1,FILE2,FILE3
*APROG.COM=BPROG.COM
*A:=B:X.A86
*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 tail is included in the command line. PIP is not loaded into memory for each command.

### 5.7.6 Using Options with PIP

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

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

Table 5-2. PIP Options

| Option | Function   |
|--------|--|
| Dn     | Delete any characters past column n. This parameter follows a source file that contains lines too long to be handled by the destination device, for example, an 80-character printer or narrow console. The number n should be the maximum column width of the destination device.   |
| E      | Echo transfer at console. When this parameter follows a source name, PIP displays the source data at the console as the copy is taking place. The source must contain character data.  |
| F      | Filter form-feeds. When this parameter follows a source name, PIP removes all form-feeds embedded in the source data. To change form-feeds set for one page length in the source file to another page length in the destination file, use the F command to delete the old form-feeds and a P command to simultaneously add new form-feeds to the destination file. |
| Gn     | Get source from or Go to user number n. When this parameter follows a source name, PIP searches the directory of user number n for the source file. When it follows the destination name, PIP places the destination file in the user number specified by n. The number must be in the range 0 to 15.  |
| H      | Hex data transfer. PIP checks all data for proper Intel® hexadecimal file format. The console displays error messages when errors occur. This function is included for historical reasons only. This PIP option does not affect Motorola S-records.  |



Table 5-2. (continued)

| Option | Function   |
|--------|--|
| I      | Ignore :00 records in the transfer of Intel hexadecimal format file. The I option automatically sets the H option. This function is included for historical reasons only. This PIP option does not affect Motorola S-records.  |
| L      | Translate upper-case alphabetic in the source file to lower-case in the destination file. This parameter follows the source device or filename.  |
| N      | Add line numbers to the destination file. When this parameter follows the source filename, PIP adds a line number to each line copied, starting with 1 and incrementing by one. A colon follows the line number. If N2 is specified, PIP adds leading zeroes to the line number and inserts a tab after the number. If the T parameter is also set, PIP expands the tab. |
| O      | Object file transfer for machine code (noncharacter and therefore nonprintable) files. PIP ignores any CTRL-Z end-of-file during concatenation and transfer. Use this option if you are combining object code files.   |

Table 5-2. (continued)

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

Table 5-2. (continued)

| Option | Function  |
|--------|---|
| V      | Verify 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      | Write over files with RO (Read-Only) attribute. Normally, if a PIP command tail includes an existing RO file as a destination, PIP sends a query to the console to make sure you want to write over the existing file. When this parameter follows a source name, PIP overwrites the RO file without a console exchange. If the command tail contains multiple source files, this parameter need follow only the last file in the list. |
| Z      | Zero the parity bit. When this parameter follows a source name, PIP sets the parity bit of each data byte in the destination file to zero. The source must contain character data.  |

Examples:

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.A86[D80]

This command writes the character file WIDEFIL.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 [NT8U]**

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

**A>PIP**

**\*PORTION.TXT=LETTER.TXT[SDear Sir^Z QSincerely^Z]**

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

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

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