

PCPM11UG.WS4 (= Personal CP/M User's Guide, version 1.1)

- "Personal CP/M Version 1.1 -- User's Guide"

(Retyped by Emmanuel ROCHE.)

Notice to user

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

Foreword

Welcome to the world of microcomputers opened to you by your 8-bit microprocessor. Welcome also to the world of application software accessible with your Digital Research Personal CP/M operating system. Digital Research designed Personal CP/M especially for the Zilog Z-80 or equivalent microprocessor that is the heart of your computer.

What Personal CP/M does for you

Personal CP/M 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. Personal CP/M can copy files from a disk to your computer's memory, or to a peripheral device such as a printer. To do this, Personal CP/M places various programs in memory, and executes them in response to commands that you enter at your console.

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

Distribution of Personal CP/M

Your Personal CP/M operating system can be contained in ROM (fixed in your system, or on a plug-in capsule), or it can be contained on floppy disk. This depends entirely on how the manufacturer of your computer has decided to deliver it. To find out how your Personal CP/M has been delivered to you, read the instructions supplied by the manufacturer of your computer.

How this guide is organized

This guide begins with simple examples, proceeds with basic concepts, then presents a detailed reference section on commands. The first 4 sections describe Personal CP/M operation for the first-time user. Section 1 introduces Personal CP/M, and tells you how to start the operating system, enter commands and edit the command line. Section 2 explains files, disks, and drives. Section 3 describes how Personal CP/M manages your printer and console. Section 4 develops the concepts you need to use Personal CP/M commands. If you are new to CP/M, read the first 4 sections carefully for a general understanding of how to use Personal CP/M, before you go on to the specific command descriptions.

Section 5 provides detailed information on each Personal CP/M utility programs, arranged alphabetically. You will not use many of these programming utilities until you start writing your own Personal CP/M programs. Section 6 tells you how to use ED, the Personal CP/M file editor. With ED, you can create and edit program source codes, text, and some data files.

Section 7 discusses ASM operation and the various assembler options which may

be enabled when invoking ASM.

Section 8 discusses the DDT program, which allows the user to test and debug programs interactively in the CP/M environment. Section 8 includes a DDT sample debugging session.

Appendix A lists Personal CP/M error messages, and describes corrective action where necessary. Appendix B provides an ASCII to hexadecimal conversion table. Appendix C lists the filetypes associated with Personal CP/M. Appendix D lists and defines the Personal CP/M control characters. This guide concludes with a glossary of commonly-used computer terms.

If you are new to computers, you might find some of the topics, such as the programming utilities, difficult to understand at first. Learning to use your computer is a challenge, and we hope that you will find it fun. This manual proceeds step by step, so you can quickly proceed from setting up your system to mastering Personal CP/M's powerful facilities.

Table of Contents

(To be done by WS4...)

Tables

(To be done by WS4...)

Section 1: Introduction to Personal CP/M

This section tells you how to start Personal CP/M. It describes the command line, and tells you how to edit it; it tells you why you should back your files.

How to start Personal CP/M

If your computer's manufacturer has built Personal CP/M into your system, when you turn ON your computer, Personal CP/M starts immediately. If your computer's manufacturer did not build Personal CP/M into the system, you must start Personal CP/M from your system disk, ROM, or cartridge -- or from the device or media on which your computer manufacturer delivered Personal CP/M.

Starting or loading Personal CP/M means reading Personal CP/M from the device or media supplied with your computer into your computer's main memory.

In the following discussion, it is assumed that Personal CP/M is supplied to you on disk. Consult the manual for your specific computer to determine the exact form in which Personal CP/M has been supplied to you.

First, check that your computer's power is ON. Next, insert the Personal CP/M 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, restart your system. (In many cases, this means pressing the RESET or RESTART button. But consult the manual for your computer to be sure. Your manuals might also refer to restarting as "resetting", "cold booting", "cold starting", or simply "loading the system".) This automatically loads Personal CP/M into memory.

After Personal CP/M is loaded into memory, a message similar to the following is displayed on your screen:

```
Personal CP/M Version 1.0 for Sharp MZ-800
Copyright (C) 1984 Digital Research Inc. / Sharp Corporation
```

Note: The "Visual CCP" is a visually-oriented version of CP/M's Console Command Processor (CCP) that prompts for commands and parameters -- much as menu-driven programs prompt for names of programs and data to be entered. If the Visual CCP automatically loads with your system, the screen formats, and the manner of entering command lines, are different, and will be documented in the manufacturer's documentation.

The version number, represented by "1.0", identifies the version of Personal CP/M that you own. After this display, the following 2-character message appears on your screen:

```
A>
```

This is the Personal CP/M system prompt. The system prompt tells you that Personal CP/M 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 Personal CP/M to do otherwise, it looks for program and data files on the disk in drive A. Also, the absence of a user number tells you that you are logged in as user zero.

The command line

Personal CP/M performs tasks according to commands that you type at your keyboard. A Personal CP/M command line is composed of a command keyword, an optional command tail, and a Carriage Return keystroke. The command keyword identifies a command (program) to execute. The command tail can contain extra information for the command, such as a filename or parameters. To end the command tail, press the Carriage Return or RETURN key. The following example shows a command line entered by a user.

A>dir myfile

In this manual, the characters that you type are shown in boldface, to distinguish them from characters that the system displays. In this example, DIR is the command keyword, and MYFILE is the command tail. The Carriage Return keystroke does not appear on the screen or in the example. You must remember to press the Carriage Return key to send a command line to Personal CP/M for processing. Note that the Carriage Return key might be marked ENTER, RETURN, CR, or something similar on your keyboard. In this guide, RETURN signifies the Carriage 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 error, press the BACKSPACE key (if your keyboard has one) or "<--" 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 Control key and press the required letter key. For example, to move the cursor to the left, hold down "Ctrl" and press the H key.

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

You type a command line directly after the system prompt. However, Personal CP/M does allow spaces between the prompt and the command keyword.

Personal CP/M recognizes 2 types of commands: built-in commands, and transient utility commands. Built-in commands execute programs that reside in memory as part of the Personal CP/M operating system. Personal CP/M executes built-in commands immediately. Transient utility commands are stored on disk as program files. They must be loaded from disk to perform their task. You can recognize transient utility program files when a directory is displayed on the screen, because their filenames are followed by COM. Section 4 includes lists of the Personal CP/M built-in and transient program commands.

For transient programs, Personal CP/M checks only the command keyword. If you include a command tail, Personal CP/M passes it to the utility without checking it because many utilities require unique command tails. A command tail cannot contain more than 128 characters. Personal CP/M cannot read either the command keyword or the command tail until you press the Carriage Return key.

The following command demonstrates how Personal CP/M reads command lines. The DIR (Directory) command tells Personal CP/M to display a directory of disk files on your screen. Type DIR after the system prompt, omit the command tail, and press RETURN.

A>dir

Personal CP/M responds to this command by writing the names of all the files stored on the disk in drive A (with the exception of SYS (system) files, which are explained later on). For example, if you have your Personal CP/M system disk in drive A, these filenames, among others, appear on your screen:

```
PIP      COM
STAT    COM
```

Personal CP/M recognizes only correctly-spelled command keywords. If you make a typing error and press RETURN before correcting your mistake, Personal CP/M

echoes the command line, followed by a question mark. If you type the DIR command incorrectly, as in the following example, Personal CP/M responds

```
A>dir
DJR?
```

to tell you that it cannot find the command keyword. To correct simple typing errors, use the BACKSPACE key, or hold down the "Ctrl" key and press the H key to move the cursor to the left. Personal CP/M supports other control characters that help you efficiently edit command lines. Section 3 tells how to use control characters to edit command lines and other information you enter at your console.

DIR accepts a filename as a command tail. You can use DIR with a filename, to see if a specific file is on the disk. For example, to check that the transient utility program STAT.COM is on your drive A, type

```
A>dir stat.com
```

Personal CP/M performs this task by displaying the name of the file that you specified, or the message "No File".

Be sure that you type at least one space after DIR, to separate the command keyword from the command tail. If you do not, Personal CP/M responds as follows:

```
A>dirstat.com
DIRSTAT.COM?
```

Why you should back up your files

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

Data processing professionals avoid losing programs and data by copying (backing up) 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. However, as a general rule, make a copy whenever it takes 10 to 20 times longer to re-enter the information than it takes to make the copy.

So far, we have not discussed commands that change recorded information on disks or other media. Before we do, if you received Personal CP/M on a disk, make a copy of it. To make a copy of a disk, follow the instructions provided by the manufacturer of your computer. If your system includes one or more disk drives, disk formatting and copy programs are included, along with instructions on how to run them.

Section 2: Files, disks, and drives

Your system might contain mass-storage devices other than disks. It can contain bubble, RAM, or ROM memory supplied on fixed boards in the system or in plug-in capsules. If you can access the data on such a device in the same way you can access that data on a disk (that is to say: in groups of data, often called "blocks"), then the device is said to be a disk-like device. In the following discussion, you can substitute "disk-like device" wherever the word "disk" occurs.

Personal CP/M's most important task is to access and maintain files on your disks (or disk-like devices). With Personal CP/M, you can create, read, write, copy, and erase 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 explains how to change disks, and change the default drive.

Although this section describes how files are stored on disks, you can apply the principles to files stored on any disk-like device.

What is a file?

A Personal CP/M file is a collection of related information stored on a disk. Every file must have a unique name, because Personal CP/M accesses files by name. A directory is also stored on each disk. The directory contains a list of the filenames stored on the disk, and the locations of each file on the disk.

Basically, there are two kinds of files: Program (command) files, and data files. A program file contains an executable program -- a series of instructions that the computer 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. 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, some executing programs can process a program file. For example, the copy program PIP can copy one or more program files.

How are files created?

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

How are files named?

Personal CP/M identifies every file by its unique file specification. A file specification can be a one- to eight-character filename, such as the following

MYFILE

A file specification can have 3 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 Personal CP/M that the file is stored on the disk currently in that drive. For example, if you enter

B:MYFILE

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

When you make up a filename, try to use a name that tells you something about the file's contents. For example, you might name a file containing a list of customer names for your business

CUSTOMER

As you begin to use your computer with Personal CP/M, your files will fall naturally into categories. To help you identify files belonging to the same category, Personal CP/M 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 ("."). Use three letters that indicate the file's category. For example, you might add the following filetype to the file that contains a list of customer names:

CUSTOMER.NAM

When Personal CP/M displays file specifications in response to a DIR command, it adds blanks to short filenames, so you can compare filetypes quickly. The program files Personal CP/M loads into memory from a disk have different filenames, but all have the filetype COM.

Create filenames and filetypes from letters and numbers. You must not use the following characters in filenames and filetypes, because they have special meanings for Personal CP/M:

< > . , ; : = ? [] - % | () / \

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

```
D:CUSTOMER.NAM
```

Do you have the correst drive?

When you type a file specification in a command tail without a drive specifier, Personal CP/M 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 stat.com
```

DIR looks in the directory of the disk in drive A for STAT.COM. If you have another drive, B, for example, you must tell Personal CP/M to access the disk in drive B, instead. For this reason, Personal CP/M lets you precede a filename with a drive specifier. For example, in response to the command

```
A>dir b:myfile.lib
```

Personal CP/M looks for the file MYFILE.LIB in the directory of the disk in drive B. When you give a command to Personal CP/M, 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

```
A>b:pip
```

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

To access many files on the same drive, you might find it convenient to change the default drive, so that you need not repeatedly enter a drive specifier. To change the default drive, enter the drive specifier next to the system prompt, and press RETURN. In response, Personal CP/M changes the system prompt to display the new default drive:

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

Unlike the filename and filetype, which are stored in the disk directory, the drive specifier for a file changes as you move the disk from one drive to another. So, a file has a different file specification when you move a disk from one drive to another. Section 4 details how Personal CP/M locates program and data files.

Do you have the correct user number?

Personal CP/M further identifies all files by assigning each one a user number, ranging from 0 to 15. Personal CP/M assigns the user number to a file when the file is created. User numbers allow you to separate your files into 16 file groups on one single disk.

When you use a Personal CP/M utility to create a file, the file is assigned 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 precede the drive identifier. User 0, however, is the default user number, and is not displayed in the system prompt.

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

```
A>user 3
3A>
```

Most commands can access only files that have the current user number. For example, if the current user number is 7, a DIR command displays only the

files created under user number 7.

Accessing more than one file

Certain Personal CP/M 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 is called an ambiguous file specification, and can refer to more than one file because it gives Personal CP/M a pattern to match. Personal CP/M searches the disk directory, and selects any file with a filename or filetype that matches the pattern.

The 2 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.

- 1) A "?" matches any character in a name, including a space character.
- 2) An "*" must be the last, or only, character in the filename or filetype. Personal CP/M internally replaces an "*" with "?" characters to the end of the filename or filetype.
- 3) When the filename to match is shorter than 8 characters, Personal CP/M treats the name as though it ends with spaces.
- 4) When the filetype to match is shorter than 3 characters, Personal CP/M treats the filetype as though it ends with spaces.

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

A.COM, AA.COM, AAA.COM, B.COM, A.ASM, and B.ASM

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

.	is treated as ??????????.???
?????????.???	matches all 6 names
*.COM	is treated as ??????????.COM
?????????.COM	matches the first 4 names
?.COM	matches A.COM and B.COM
?.*	is treated as ?.???
?.???	matches A.COM, B.COM, A.ASM, and B.ASM
A?.COM	matches A.COM and AA.COM
A*.COM	is treated as A?????????.COM
A?????????.COM	matches A.COM, AA.COM, and AAA.COM

Remember, Personal CP/M uses wildcard patterns only when searching a disk directory, so wildcards are valid only in filenames and filetypes. You cannot use a wildcard character in a drive specifier. Nor can you use a wildcard character as part of a filename or filetype when you create a file.

How to protect your files

Under Personal CP/M, you can organize your files into groups, to protect them from accidental change. You can also specify how your files are displayed in response to a DIR command. Personal CP/M supports these features by assigning a user number and attributes to each file.

All this information is recorded in the disk directory. File attributes control how programs access files. When you create a file, Personal CP/M gives it 2 attributes. You can change the attributes with a STAT command.

You can set the first attribute to DIR (Directory) or SYS (System). This attribute controls whether Personal CP/M displays the file's name in response to a DIR command. When you create a file, Personal CP/M automatically sets this attribute to DIR. You can display the name of a file marked with the DIR attribute with a DIR command. If you give a file the SYS attribute, the DIR command will not display the filename.

Note: To display a SYS file, use the STAT command with the command tail *.*. The DIR command displays only the filenames created under the current user number.

A file with the SYS attribute has a special advantage when it is created under user zero. When you give a file with user number 0 the SYS attribute, you can read and execute that file from any user number. This feature makes your commonly-used programs available under any user number.

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, attempting 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, written to, or erased at any time, unless the disk is physically write-protected.

How are files stored on a disk?

Personal CP/M records the filename, filetype, user number, and attributes of each file in a special area of the disk called the directory. The Personal CP/M directory also records the location of each file on the disk.

Personal CP/M allocates directory and storage space for a file as you add records to the file. When you erase a file, Personal CP/M reclaims storage in 2 ways: 1) it makes the file's directory space available to catalog a different file, and 2) it frees the file's storage space for later use. This dynamic allocation feature makes Personal CP/M powerful. You need not tell Personal CP/M how big your file will become, because Personal CP/M 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 find out how much space remains on the disk.

Changing floppy disks

Personal CP/M cannot do anything to a drive, 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 online, and Personal CP/M can access its directory and files.

At some time, you must take a disk out of a drive, and insert another that contains different files. You can replace an online disk whenever the system prompt appears on your console. The system prompt indicates that no program is reading 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 do not fit on one floppy disk.

Note: Never remove a disk while a program is reading or writing to it.

You can change disks on the drive without sending any special signals to Personal CP/M. You can insert a different disk at a program's request, and read files from, or create files on, the new disk.

Protecting a drive

Under Personal CP/M, drives can be marked R-O (Read-Only), just as files can be given the R-O attribute. The default state of a drive is R/W (Read/Write). You can give a drive the R-O attribute by using the STAT command described in Section 5. To return the drive to R/W, use the STAT command, or press Ctrl-C to return to the system prompt.

Section 3: Console and printer

This section describes how Personal CP/M communicates with your console and printer. It tells how to start and stop console and printer output, and how to edit commands you enter at your console.

Controlling console output

Sometimes, Personal CP/M 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 Personal CP/M to wait while you read the display, hold down the Control ("Ctrl") key and press S. A Ctrl-S keystroke causes the display to pause. When you are ready, press any other key to resume the display.

Controlling printer output

You can also use a control command to echo console output to the printer. To start printer echo, press Ctrl-P. To stop, press Ctrl-P again. While printer echo is in effect, 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 stored on a floppy disk. You can also use Ctrl-P with Ctrl-S to make a hard copy of part of a file. Use a TYPE command to start a display of the file at the console. When the display reaches the part you want to print, press Ctrl-S to stop the display, and Ctrl-P to enable printer echo. Then, press any key to resume the display and start printing. Use another Ctrl-S/Ctrl-P sequence to terminate printer echo.

Console line editing

You can correct simple typing errors with the BACKSPACE key. Personal CP/M 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.

Personal CP/M allows you to edit your command line using the control characters listed in Table 3-1. To edit a command line in Personal CP/M, 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 incorrectly typed. (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, type the correct letters, and press RETURN, sending the command line to Personal CP/M.

```
A>pip a:=b:*.*_
```

Table 3-1 describes Personal CP/M control characters.

Table 3-1. Personal CP/M control characters

Format: Character
Meaning

Ctrl-C
Warm boots (restarts) the Personal CP/M operating system when typed at the beginning of a line.

Ctrl-E
Forces a physical Carriage Return, but does not send the command line to Personal CP/M. 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.

Ctrl-J
Sends the command line to Personal CP/M, 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 Personal CP/M, 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 # sign at the current cursor location, moves the cursor to the next line, and displays any partial command you typed so far.

Ctrl-U
Discards all the characters in the command line (but leaves them displayed), 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 (actually, removes them from display), and moves the cursor to the beginning of the current line.

RUBOUT/DEL

Deletes the last character typed, and echoes it at the console.

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 3 send the command line to Personal CP/M for processing. Also, Ctrl-H has the same effect as pressing the BACKSPACE key.

Section 4: Personal CP/M command concepts

As explained in Section 1, a Personal CP/M command line consists of a command keyword, an optional command tail, and a Carriage Return keystroke. This section describes the 2 kinds of programs that the command keyword can identify, and tells how Personal CP/M searches for a program file on a disk. This section also explains how to execute multiple Personal CP/M commands, and how to terminate programs and reset the disk system.

Two kinds of commands

A command keyword identifies a program that resides in memory as part of Personal CP/M, 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 commands.

Personal CP/M has 6 built-in commands, and 8 transient program commands. You can add programs to your system by purchasing Personal CP/M-compatible application programs. If you are an experienced programmer, you can also write your own programs that operate with Personal CP/M.

Built-in commands

Built-in commands are part of Personal CP/M. You can always use them, regardless of which disk you have in which drive. Built-in commands reside in memory as a part of Personal CP/M, and therefore execute more quickly than the transient programs.

Section 5 explains in details the built-in commands listed in Table 4-1.

Table 4-1. Built-in commands

Command Function

DIR	Displays filenames of all files in the directory, except those marked with the SYS attribute.
ERA	Erases a filename from the disk directory, and releases storage space occupied by the file.
REN	Renames a disk file.
SAVE	Stores a portion of main memory in a disk file.
TYPE	Displays the contents of an ASCII (text) file at your screen.
USER	Allows you to change to a different user number.

Transient program commands

When you enter a command keyword that identifies a transient program, Personal CP/M loads the program file from the disk, and passes it any filenames, data, or parameters that you entered in the command tail. Section 5 provides the operating details for the Personal CP/M transient programs listed in Table 4-2. These utilities are used only by experienced programmers.

Eight other transient programs are available: COPY, DUMP, RANDOM, TERMINAL, BLTMEMO, ASM, LOAD, and DDT. COPY, DUMP, RANDOM, TERMINAL and BLTMEMO, described in the "Personal CP/M Programmer's Guide", are included in this manual as sample programs supplied in source code only. If you assemble the source code, you can run these programs. ASM (the assembler program), LOAD

(the absolute loader), and DDT (the debugger) are described in this manual.

Table 4-2. Transient program commands

Command Action

ED	Loads and executes the CP/M programmer's editor program; creates and alters character files.
PIP	Loads and executes the Peripheral Interchange Program, which copies, combines, or transfers files.
STAT	Displays statistical information, including space in kilobytes occupied by a file; file attributes; disk status (Read-Only, or Read/Write). Also allows you to set file attributes and disk status.
SUBMIT	Executes a list of commands contained in a file.
XSUB	This additional utility program is used in conjunction with SUBMIT. XSUB extends the power of the SUBMIT utility to include line input to programs, as well as to the console command processor.

How Personal CP/M searches for files

If Personal CP/M cannot find a program file that you specified in a command line, Personal CP/M might not be looking on the drive on which the file is stored. This section explains how Personal CP/M searches for program and data files.

Finding data files

When you enter a command line, Personal CP/M passes the command tail to the program identified by the command keyword. If the command tail contains a file specification, the program calls Personal CP/M to search for the data file. If Personal CP/M cannot find the data file, the program displays an error message at the console. Typically, this message is "File not found", or "No file", but the exact message depends on the program the command keyword identifies.

If you do not include a drive specifier with the filename in a command tail, Personal CP/M searches the directory of the current user number on the default drive. If the file is not there, Personal CP/M looks for the file with the SYS attribute in the directory of user 0 on the default drive. If Personal CP/M finds the file under user 0, it allows the program Read-Only access to the file. For example, if you enter the following command line:

```
3A>type myfile.ASC
```

Personal CP/M first searches the directory for user 3 on drive A. If it does not find MYFILE.ASC there, it searches the directory of user 0 on drive A for MYFILE.ASC marked with the SYS attribute. If the file is not in either directory, Personal CP/M returns control to TYPE, which then displays "No file".

Some Personal CP/M utilities, such as PIP and DIR, restrict their file search to the current user number. Because Personal CP/M does not allow Read/Write access to SYS files, ERA and REN also restrict their search to the current user number.

The search procedure is basically the same if you include a drive specifier with the filename. Personal CP/M first looks in the directory of the current user number on the specified drive. Then, if it does not find the file, it looks in the directory for user 0 on the specified drive for the file with the SYS attribute. If Personal CP/M does not find the data file after these 2 searches, it displays an error message.

Finding program files

If a command keyword identifies a transient program, Personal CP/M looks for that program file on the default or specified drive. It looks under the current user number, and then under user 0 for the same file marked with the SYS attribute. At any point in the search process, Personal CP/M stops the search if it finds the program file. Personal CP/M then loads the program into memory, and executes it. When a program terminates, Personal CP/M displays the system prompt, and waits for your next command. However, if Personal CP/M does not find the command file, it repeats the command line, followed by a question

mark, and waits for your next command.

When you include a drive specifier before the command keyword, you tell Personal CP/M to look on that drive for the program file. Personal CP/M then searches 2 locations: the directory for the current user on the specified drive, and then for user 0 on the specified drive, before it repeats the command line with a question mark. For example, if you enter

```
4B>a:stat space
```

Personal CP/M looks on drive A, user 4, and then user 0 for the file STAT.COM.

Executing multiple commands

Personal CP/M can execute a sequence of commands. You can put a frequently needed sequence of commands into a disk file. Once you have stored the sequence in a disk file, you can execute the entire sequence with a single SUBMIT command.

Store command sequences that you execute frequently in a disk file. To create this file, use ED or another character file editor. The file must have a filetype of SUB, and each command in the file must start on a new line. For example, an UPDATE.SUB file might look like this:

```
DIR A:*.COM
ERA B:*.COM
PIP B:=A:*.COM
```

To execute this list, enter the following command:

```
A>submit update
```

The SUBMIT utility passes each command to Personal CP/M for sequential execution. While SUBMIT executes, the commands usually echo at the console. When one command completes, the system prompt reappears with the next command in the SUB file, or else reappears by itself, when the SUB file is exhausted. SUBMIT then waits for your next command from the keyboard.

The SUBMIT command is detailed in Section 5.

Terminating programs

The Ctrl-C keystroke terminates program execution, or resets the disk system. To enter a Ctrl-C command, hold down the "Ctrl" key and press the C key.

Not all application programs that run under CP/M terminate with a Ctrl-C. However, you can terminate most of the transient programs supplied with Personal CP/M immediately with a Ctrl-C keystroke. If you want to terminate a program that is sending a display to the screen, you might have to press Ctrl-S to halt the display before entering Ctrl-C.

Ctrl-C also resets the disk system. This is called a warm boot or warm start. When you press Ctrl-C and the cursor is at the system prompt, Personal CP/M logs out all the active drives, then logs in the default drive. The active drives are any drives that you have accessed since last cold or warm start. A STAT command displays the remaining space on all active drives. In the following example, STAT indicates that 3 drives are active. However, if you press Ctrl-C immediately after this display, and then enter another STAT command, only the space for the default drive, A, is displayed.

```
A>stat
A: R/W, Space: 9,488k
B: R-O, Space: 2,454k
C: R-O, Space: 1,665k
A>^C
A>stat
A: R/W, Space: 9,488k
```

Section 5: Command summary

This section describes the commands and programs supplied with your Personal CP/M operating system. The commands are listed alphabetically, along with short explanations and examples.

ED is described in greater detail in Section 6. ASM (the assembler), LOAD (the absolute loader), and DDT (the debugger) are described in the "Personal CP/M Programmer's Guide". Five other programs -- COPY, DUMP, RANDOM, TERMINAL, and BLTMEMO -- are supplied only in source code. See the "Personal CP/M Programmer's Guide" for their descriptions.

Let's get past the formalities

This paragraph describes the parts of a file specification in a command line. A file description names a file or group of files in the directory of the online 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 "filespec" in the command syntax statements. The 3 parts of a file specification are:

- 1) The drive specifier -- the optional disk drive A, B, C through P, that contains the file or group of files to which you refer. If you include a drive specifier in your command line, a colon (":") must follow it.
- 2) The filename -- the one- to eight-character name of a file, or group of files.
- 3) The filetype -- the optional one- to three-character category name of a file, or group of files. A period (".") must separate the filetype from the filename.

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

Some Personal CP/M 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 ambiguous file specification might match several files in the directory of drive B.

Put together, the parts of a file specification are represented in the following general form:

x:filename.typ

In the preceding form, X 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 Personal CP/M file specification.

- filename
- filename.typ
- x:filename
- x:filename.typ

The following characters have special meaning in Personal CP/M, so do not use them, except as specified in a description:

< > . , ; : = ? [] - % | () / \

Personal CP/M has established several file groups. Table 5-1 lists some of their filetypes, with a short description of each family. Appendix C provides the complete list.

Table 5-1. Personal CP/M filetypes

Filetype	Meaning
ASM	Assembler source file
BAS	BASIC source program
COM	Machine language program
HLP	HELP message file
SUB	List of commands to be executed by SUBMIT
\$\$\$	Temporary file

In some commands, descriptive qualifiers are used with filespecs to further define 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.

You now understand command keywords, command tails, control characters, default drives, and wildcards. You also see how to use the formal names filespec, drive specifier, filename, and filetype. These concepts give you the background necessary to compose complete command lines.

How commands are described

Personal CP/M commands are presented in alphabetical order by command keyword. The command description format is as follows:

- 1) The command keyword appears in uppercase.
- 2) The syntax section gives you one or more general form to follow when you compose the command line.
- 3) The explanation section defines the command keyword, and points out exceptions and special cases. Some explanations include tables or lists of options that you can use in the command line.
- 4) The examples section lists a number of valid command lines. To clarify examples of interactions between you and the operating system, the characters that you enter on your console are shown in boldface.

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

- 1) Words in capital letters must be spelled as shown, but you can use any combination of upper- or lower-case letters.
- 2) Words italicized in the syntax line are defined in the text.
- 3) The symbolic notation x:, filename, typ, and filespec have the general meanings described earlier in this section.
- 4) You must include one or more space characters where a space is shown, unless otherwise specified. For example, the PIP options need not be separated by spaces.

Table 5-2 defines the special symbols and abbreviations used in syntax lines.

Table 5-2. Syntax notation

Symbol	Meaning
DIR	Directory attribute.
n	You can substitute a number for n.
o	Indicates an option, or an option list.
R-O	Read-Only. Indicates a file or disk that can only be read.
R/W	Read/Write. Indicates a file that can be read and written.
s	You can substitute a string, which consists of a group of characters, for s.
SYS	System attribute.
{ }	Items within curly brackets are optional. You can enter a command without optional items. The optional items add effects to your command line.
[]	Items in square brackets are options, or an option list. When you use an option specified within the square brackets, you must enclose the option in square brackets. If the right-closing square bracket ("]") is the last character on the command line, you can omit it.
()	Items in parentheses indicate a range of options. If you use a range from an option list, you must enclose the range in parentheses.
...	The item preceding the ellipses 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 selected. Mutually exclusive options are indicated in additional syntax lines, or are specifically noted in the text.

Ctrl Represents the Control ("Ctrl") key on your keyboard.
or ^ (Control characters appear as ^char on your screen.)

Indicates a Carriage Return keystroke.

* Wildcard character -- any valid group of characters can take the place of the *.

? Wildcard character -- any valid character can take the place of the ?.

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

The syntax of the DIR command is

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

The curly brackets indicate that the command tail following the command keyword DIR is optional. DIR alone is a valid command, but you can include a file specification, a drive specifier, or both. Thus, the following forms of the DIR command are valid:

```
DIR
DIR x:
DIR filename
DIR filename.typ
DIR x:filename.typ
```

Recall that, in Section 2, you learned about wildcards in filenames and filetypes. The DIR command accepts wildcards in the file specification. So, command lines like the following are valid:

```
DIR B:*.C?M
```

The Personal CP/M command PIP (Peripheral Interchange Program) calls the file copy program. PIP copies information from the disk to the screen or printer. PIP combines two or more files into one longer file. PIP also renames file after copying them, and copies files from disk to disk. Look at one of the formats of the PIP command line for another example of command line notation.

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

In the preceding 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 description for other capabilities of PIP.) Many valid command lines come from this syntax. Some examples follow:

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

A complete description of each Personal CP/M utility follows. The descriptions are arranged alphabetically.

DIR Command

```
Syntax: DIR {x:} {filename.typ}
```

Explanation:

The DIR (Directory) command displays the names of all directory files in the current user number denoted by the drive and file specifications.

The drive and the FILENAME.TYP specifications are optional; either or both can appear. Both filename and filetype can contain wildcard characters. If no

drive specifier is set, DIR assumes the currently-logged drive. If you omit the FILENAME.TYP specification, DIR displays the names of all files with the DIR attribute on the currently-logged or specified drive. DIR by itself is equivalent to

```
DIR *.*
```

where the drive is the currently-logged drive.

The DIR command displays only files with the DIR attribute. Use the STAT command to display files with the SYS attribute. Only files under the current user number are displayed.

If no file meets the drive/file/user specifications, DIR displays the message

```
NO FILE
```

Examples:

```
A>dir
```

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

```
A>dir b:
```

Displays all DIR files in user 0 on drive B.

```
4B>dir *.nam
```

For user 4 on drive B, displays all DIR files with filetype NAM.

```
4B>dir c:*.*
```

For user 4 on drive C, displays all DIR files.

ED Command

Syntax: ED {x1:}filename.typ {x2:}

Explanation:

ED allows you to create and edit disk files.

Drive specifications X1 and X2 (both standing for drives A through P) are optional. When X1 appears, Personal CP/M looks for the source file on drive X1. When X2 appears, Personal CP/M places the temporary edit file (used during actual file editing) on drive X2. When editing is complete, the edited file appears on drive X2. The temporary edit file is named FILENAME.***. When editing is complete, the old source file, named FILENAME.TYP, is renamed FILENAME.BAK, and the temporary edit file (FILENAME.***) is renamed FILENAME.TYP.

The filename and filetype cannot contain wildcard characters; the filetype is optional.

ED uses a portion of main memory as a buffer for sections of text being edited. You move text into this buffer with the A (Append) command. You write text from the buffer to the temporary edit file (FILENAME.***) with the W (Write) command. The E (Exit) command functions like the W command. It also renames FILENAME.TYP to FILENAME.BAK, and FILENAME.*** to FILENAME.TYP, and returns to the Personal CP/M prompt. (A full command list for ED appears in Section 6.)

If no file named FILENAME.TYP exists when you type

```
A>ed filename.typ
```

ED displays the message

```
NEW FILE
```

and opens a file for you to edit.

You interact with the ED utility in command or insert mode. ED displays the prompt "*" on the screen when ED is in command mode, and you can enter a variety of command characters (described in Section 6) that allow you to manipulate text. In insert mode, you can insert new text into the file. In insert mode, line numbers "nnnnn" (5 digits) appear at the beginning of each line. These numbers are displayed for reference only, and are not contained in the buffer, or in any of the disk files. You can disable line numbers with the -V command.

Examples:

A>ed x.asc

Creates a temporary work file named X.***. Allows you to edit source file in a buffer in main memory, and store edited file in the temporary file. At completion of editing, source is renamed X.BAK, and X.*** is renamed X.ASC. All operations occur on the currently-logged disk.

A>ed c:z.asc b:

Creates a temporary work file named Z.*** on drive B, allowing you to edit source in a buffer in main memory. At completion of editing, ED renames Z.ASC on drive C Z.BAK, and renames Z.*** on drive B Z.ASC.

See Section 6 for a complete description of ED.

ERA Command

Syntax: ERA {x:}filename.typ

Explanation:

This command erases files from a disk.

The drive specifier X (A through P) is optional. If no drive specifier is given, Personal CP/M deletes the file, or files, from the currently-logged disk. The FILENAME.TYP can contain wildcard characters. The TYP is optional.

If no file, or set of files, matches FILENAME.TYP, the following message appears:

NO FILE

Use ERA with care, because ERA erases every file that matches FILENAME.TYP. When using wildcards in the file specification, be sure that you mean to delete all files denoted by the file specification. (To be sure, you could first check the result of your file specification, using the DIR command, before giving your ERA command.)

This command takes place in the currently-logged user number. Directory and data space are immediately reclaimed for use by other files.

Examples:

2B>era x.asc

Erases the file X.ASC in user 2 of drive B.

3C>era *.asc

Erases all files with filetype ASC in user 3 of drive C.

PIP Command

Syntax: PIP
PIP 'command line'

Explanation:

PIP is a transient program that 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 output device. PIP can create a file on disk from input from the console or other auxiliary input device. PIP can transfer data from a logical input device to a logical output device, thus the name "Peripheral Interchange Program" (PIP).

To initiate PIP, type one of the preceding forms. Both forms load PIP into the TPA and execute it. When you use the first form, PIP reads command lines directly from the console, prompting you with the "*" character, until you enter an empty command (that is to say: until you press RETURN, without typing anything). Each successive command line causes a media conversion to take place. The form PIP 'command line' is equivalent to the form PIP, except that the single command line you type with PIP automatically executes, and PIP terminates immediately. The form of each command line is

destination = source1, source2, ..., sourceN

where DESTINATION is the file or peripheral device to receive the data, and SOURCE1, ..., SOURCEN is a series of files or devices that are copied from left to right to the destination.

When the command line specifies multiple files (that is to say: when N is

greater than 1 in the preceding form), PIP assumes that the files contain ASCII characters with a CP/M end-of-file character (Ctrl-Z, 1AH) at the end of each file. (See the O parameter to override this assumption.) PIP internally translates lowercase ASCII alphabets to uppercase for consistency with CP/M file and device name conventions. Finally, the total command line length must not exceed 128 characters. Use Ctrl-E to force a physical Carriage Return for lines that exceed 128 characters.

The destination and source elements are unambiguous references to CP/M source files with, or without, a preceding drive specifier. You can reference any file with a preceding drive specifier (A through P) that defines the drive on which the file is stored. When you do not include a drive, PIP assumes the currently-logged disk. The destination file can also appear as one or more source files; in this case, PIP does not alter the source file until the entire concatenation is complete. If the destination file already exists, it is removed if the command line is properly formed.

The destination file is not removed if an error condition arises.

Examples:

A>**pip x=y**

Copies to file X from file Y, where X and Y are unambiguous filenames; Y remains unchanged.

A>**pip x.asm=y.asm,z.asm,fin.asm**

Creates the file X.ASM from the concatenation of the Y, Z, and FIN files with filetype ASM.

A>**pip b:a.u=b:b.v,a:c.w,d.x**

Concatenates file B.V from drive B with C.W from drive A and D.X from the logged disk; creates the file A.U on drive B.

With abbreviations

Syntax: Destination=Source

```
PIP x:=filename.typ      (wildcards allowed)
PIP x1:=x2:filename.typ  (wildcards allowed)
PIP filename.typ=x2:     (wildcards not allowed)
PIP x1:filename.typ=x2:  (wildcards not allowed)
```

PIP allows the foregoing abbreviated commands for transferring files between disk drives.

The first form copies all files that satisfy FILENAME.TYP from the currently-logged disk to the same files on drive X (A through P). The second form copies all files that satisfy FILENAME.TYP from drive X2 (the second drive specifier) to the same files on drive X1 (the first drive specifier). The third form copies all files that satisfy FILENAME.TYP on drive X2 to the same files on the currently-logged disk. The fourth form copies all files that satisfy FILENAME.TYP on drive X2 to the same files on drive X1.

The source and destination disks must be different in all these cases. An ambiguous filename is a filetype that contains wildcard characters. An unambiguous filename is a filename that contains no wildcard characters. If an ambiguous filename is specified, PIP lists each unambiguous filename that satisfies that ambiguous filename as it is being copied. If a file of the same name as the destination file exists, it is removed on successful completion of the copy, and replaced by the copied file.

Examples:

A>**pip b:=*.com**

Copy all files that have filetype COM to drive B from the current drive.

A>**pip a:=b:zap.***

Copy all files that have the filename ZAP to drive A from drive B.

A>**pip zap.asm=b:**

Equivalent to ZAP.ASM=B:ZAP.ASM

A>**pip b:zot.com=a:**

Equivalent to B:ZOT.COM=A:ZOT.COM

Command line with disk or character devices

Syntax: device0:filename0.typ0=
 device1:filename1.typ1, device2:filename2.typ2, ...
 deviceN:filenameN.typN

Explanation:

PIP allows reference to the various character devices attached to the CP/M system. The character devices that you can reference are listed in Table 5-3.

Table 5-3. Personal CP/M character devices

Device	Explanation
CON:	Console. Input/Output device.
LST:	List device (printer). Output device.
PRN:	Same as LST: with options [t8np60]. Expands tabs to 8 spaces, numbers the lines, and prints 60 lines on a page before sending an ASCII Form-Fed (Ctrl-L, 0CH) to the printer.
EOF:	End of file. Sends a CP/M end-of-file character (Ctrl-Z, 1AH) to the destination device. Input device.
NUL:	Null device. Sends 40 ASCII NULL characters (Ctrl-@, 00H) to the destination device. Input device.
INP:	Input device. Special input device that can be patched into the PIP program by your computer manufacturer. See the user's manual for your computer for more information about this device. Input device.
OUT:	Output device. Special output device that can be patched into the PIP program by your computer manufacturer. See the user's manual for your computer for more information about this device. Output device.
AUX:	Auxiliary device. Typically used for a serial I/O device such as a IEEE RS-232C interface. Input/Output device.

Note that the destination device must be capable of receiving data, and the source devices must be capable of generating data. For example, the LST: device cannot be read.

You can intersperse file and device names in the PIP commands. In each case, PIP reads the specific device until end-of-file (a Ctrl-Z character for ASCII files, and an end-of-data condition for disk files). PIP concatenates data from each device or file from left to right, until the last data source has been read. The destination device or file is written using the data from the source files, and an end-of-file character (Ctrl-Z, 1AH) is appended to the result for ASCII files. If the destination is a disk file, PIP creates a temporary file (filetype = \$\$\$) that changes to the actual filetype only on successful completion of the copy. PIP assumes files with the type COM to be non-ASCII.

You can abort a copy operation in 2 ways: If you press Ctrl-Z, PIP treats this like an end-of-file from the device; that is to say: it treats it like a normal termination. If you press any other key, the operation aborts and the message "ABORTED" appears on the console. If an operation aborts, or if an error occurs during processing, PIP removes pending commands that were set up while using the SUBMIT command.

PIP performs a special function if the destination is a disk file with type HEX (a file containing machine code in Intel HEX file format), and the source is an external peripheral device, such as a paper tape reader. In this case, PIP checks to ensure that the source file contains a properly-formed HEX file, with legal hexadecimal values and checksum records.

When PIP finds an invalid input record, PIP reports an error message at the console, and waits for corrective action. It is usually sufficient to open the paper tape reader and re-run a section of the tape (pull the paper tape back about 20 inches). When the paper tape is ready for the re-read, press RETURN once, and PIP attempts another read. If PIP cannot read the paper tape position properly, continue the read by pressing RETURN following the error message. Enter the record manually with the ED program after the disk file is constructed. PIP allows you to enter the end-of-file from the console. As noted above, when you type Ctrl-Z at the keyboard, the read operation terminates normally.

Examples:

A>pip lst:=x.prn

Copies X.PRN to the list device, and terminates the PIP program.

A>pip con:=x.asm,y.asm,a.asm

Concatenates 3 ASM files on the currently-logged disk, and compiles them to the console device.

Command line with parameters

Syntax: source n{[parameters]}

You can also specify one or more PIP parameters. Enclose the parameters in square brackets, and separate them with zero or more blanks. Each parameter affects the copy operation, and the enclosed list of parameters must immediately follow the affected file or device. An optional decimal integer value typically follows each parameter. (The S and Q parameters are exceptions.)

Valid PIP parameters are listed in Table 5-4.

Table 5-4. Valid PIP parameters

Format: Parameter
Explanation

B
Block mode transfer: PIP buffers data until a XOFF character (Ctrl-S, 13H) is received by the source device. This allows transfer of data to a disk file from a continuous reading device, such as a magnetic tape reader. Upon receipt of the XOFF, PIP clears the disk buffers, and returns for more input data. The amount of data that can be buffered depends on the memory size of the host system. PIP issues an error message when the buffer overflows.

Dn
Delete characters that extend past column n in the transfer of data to the destination from the character source. This parameter truncates long lines that are sent to a (narrow) printer or console device.

E
Echoes all transfer operations to the console as they are being performed.

F
Filters ASCII Form-Feeds characters (Ctrl-L, 0CH) from the file. All imbedded Form-Feeds are removed. The P parameter can simultaneously insert new Form-Feeds.

Gn
Get file from user number n (n in the range 0 to 15).

H
HEX file data transfer. Checks all data for proper Intel HEX file format. Removes non-essential characters between hex records during the copy operation. The console prompts for corrective action in case errors occur.

I
Ignores :00 records during the transfer of Intel HEX files. (The I parameter automatically sets the H parameter.)

L
Translates uppercase alphabets to lowercase.

N
Adds line numbers to each line transferred to the destination, starting at 1 and incrementing by 1. Leading zeroes are suppressed, and a colon (":") follows the number. If "N2" is specified, leading zeroes are included, and an ASCII tab character (Ctrl-I, 09H) is inserted following the number. The tab is expanded if T is set.

O
Object (non-ASCII) file transfer: the CP/M end-of-file character is ignored.

Pn
Includes page ejects at every n lines (with an initial page eject). If n = 1, or is excluded, page ejects occur every 60 lines. If you use the F parameter, Pn suppresses Form-Feed before the new page ejects are inserted.

Qs^z
Quits copying from the source device or file when the string s (terminated by a Ctrl-Z, hence the "^z" above) is encountered.

R
Reads system files.

Ss^z
Starts copying from the source device when the string s (terminated by a Ctrl-z, hence the "^z" above) is encountered. You can use the S and Q parameters to abstract a section of a file, such as a subroutine. The copy operation always includes the start and quit strings.

If you select the PIP 'command line' form, the CCP translates strings following the S and Q parameters to uppercase. The command form PIP does not perform the automatic uppercase translation.

Tn
Expands ASCII tab (Ctrl-I, 09H) characters to every nth column during the transfer of characters to the destination from the source.

U
Translates lowercase alphabets to uppercase during the copy operation.

V
Verifies that data has been copied correctly, by re-reading after the write operation.

W
Writes over Read-Only files, without console interrogation.

Z
Zeroes the parity bit from each ASCII character.

Examples:

A>**pip x.asm=b:[v]**
Copies X.ASM from drive B to the current drive, and verifies that the data were properly copied.

A>**pip lst:=x.asm[nt8u]**
Copies X.ASM to the list device, numbers each line, expands tabs to every eight column, and translates lowercase alphabets to uppercase.

A>**pip lx.lib=y.asm[ssubri:^zqjmp L3^z]**
Copies from the file Y.ASM into the file X.LIB. Starts the copy when PIP finds the string "SUBRI:", and quits copying when the string "JMP L3" is encountered.

A>**pip prn:=x.asm[p50]**
Sends X.ASM to the list device with line numbers, tabs expanded to every eight column, and page ejects at every 50th line. The assumed parameter list for a PRN file is nt8p60; p50 overrides the default value.

A>**pip a:=b:*.com[w]**
A>**pip a.dat=b.dat,f:new.dat,g:old.dat[w]**
PIP does not overwrite a file set to a permanent Read-Only status. If you attempt to overwrite a read-Only file, PIP responds

DESTINATION FILE IS R-O, DELETE (Y/N) ?

If you press the character Y, the file is overwritten. Otherwise, PIP responds

** NOT DELETED **

PIP skips the file transfer, and continues with the next operation in sequence. To avoid the prompt, and response for Read-Only file overwrite, include the W parameter in the command line, shown in the 2 preceding examples. In the first example, PIP copies all non-system COM files from drive B to drive A, overwriting any Read-Only files in the process. If the operation involves several concatenated files, you must include the W parameter only with the last file in the list, as shown in the second example.

REN Command

Syntax: REN {x;}newfile.typ={x;}oldfile.typ

Explanation:

The REN (Rename) command allows you to change the names of files on disk.

OLDFILE.TYP changes to NEWFILE.TYP. The drive specifier X (A through P) is

optional in the command. It can precede NEWFILE.TYP or OLDFILE.TYP, or both. If the drive specifier precedes both, then X must reference the same drive in both cases. If X precedes only one of the filenames, REN assumes that the renaming operation takes place on that drive. If no drive specifier is specified, then the REN command assumes the currently-logged drive.

The filename and filetype must not contain wildcards. The filetype is optional.

If NEWFILE.TYP is already present, REN responds with the error message

```
FILE EXISTS
```

and makes no change. If OLDFILE.TYP does not exist on the specified disk, the message

```
NO FILE
```

prints at the console.

Examples:

```
A>ren x=y
```

Changes the name of file X to Y.

```
A>ren b:two.asm=one.asm
```

Changes the name of file ONE.ASM to TWO.ASM on drive B.

SAVE Command

Syntax: SAVE n {x:}filename.typ

Explanation:

The SAVE command allows you to save a portion of memory in a disk file. The SAVE command writes n pages (256-byte blocks) of the TPA (Transient Program Area) of Personal CP/M to a file on drive X, and names the file FILENAME.TYP.

The drive specifier X (A through P) is optional. If no drive specifier is specified, the currently-logged drive is assumed. The filetype is also optional.

The TPA starts at location 0100H (=256 in decimal). If your program occupies memory locations 0100H through 02FFH (256 through 767), you must specify 2 pages in order to save it ($256 + 2 \times 256 - 1 = 767$).

Examples:

```
A>save 3 x
```

Copies memory locations 0100H through 03FFH to a file named X on the currently-logged drive.

```
A>save 20 c:x.sav
```

Copies memory locations 0100H through 15FFH to a file named X.SAV on drive C.

STAT Command

Syntax: STAT
STAT 'command line'

Explanation:

The STAT command provides general status information about file storage and drive. Initiate STAT by typing one of the preceding command forms.

Special forms of the command line allows you to examine and alter the current device assignment. The various command lines that you can specify are shown and explained below.

Examples:

```
A>stat
```

When you type an empty command line, the STAT transient calculates the storage remaining on all active drives, and prints a message.

```
x: R/W, SPACE: nnnK
```

or

```
x: R-O, SPACE: nnnK
```

for each active drive X (A through P), where "R/W" indicates that the drive can be read or written, and "R-O" indicates that the drive is Read-Only (a drive becomes Read-Only by setting it to To Read-Only, as shown below). The space remaining on the disk in drive X, in kilobytes, is denoted by nnn.

Display remaining bytes remaining on drive

Syntax: STAT x:

Explanation:

When you specify a drive specifier X (A through P), STAT selects the drive before computing the storage. Thus, if you issue the command STAT B: while logged into drive A, the following message results:

```
BYTES REMAINING ON B: nnnK
```

Specify files

Syntax: STAT filename.typ

Explanation:

The filename and filetype can contain wildcard characters.

The command line can also specify a set of files for STAT to scan. The files that satisfy the filename and filetype specified are listed in alphabetical order, with storage requirements for each file under the heading

```
RECS  BYTS  EX  X:FILENAME.TYP
rrrr  bbbK  ee  x:filename.typ
```

where RECS is the number of 128-byte records allocated to the file; BYTS is the number of kilobytes allocated to the file (bbb = rrrr * 128 / 1024); EX is the number of 16-kilobyte extents (ee = bbb / 16); X is the drive containing the file (A through P), FILENAME is the name of the file (up to 8 characters long); and TYP is the type of the file, which can be up to 3 characters long. After listing the individual files, STAT summarizes the storage usage.

Specify drive and files

Syntax: STAT x:filename.typ

Explanation:

This syntax gives the drive specifier X (A through P) before the filename and filetype. STAT selects the specified drive, then executes.

Set drive status

Syntax: STAT x:=R-O

Explanation:

This form set drive X (A through P) to Read-Only. Read-Only status remains in effect until the next warm or cold start. When a drive is set for read-Only, the message

```
CP/M Error on x: Read-Only Disk <---- Drive ?
```

appears when you try to write to the Read-Only drive X. Personal CP/M waits until you depress a key before performing an automatic warm start, at which time the drive becomes Read/Write.

For available status commands

Syntax: STAT HELP

Explanation:

This command produces an instant summary of the possible STAT commands, output as follows:

```
Temp R-O Drive:  x:=R-O
Set  Attribute:  filename.typ [R-O [R/W [SYS [DIR
```

Disk Status: DSK: x:DSK:
User Status: USR:

Display file information

Syntax: STAT x:filename.typ [S

Explanation:

X is an optional drive specifier (A through P); the filename and filetype can contain wildcard characters. This form of the STAT command produces the output display format:

Size	Recs	Bytes	Ext	Att
48	55	6k	1	R-O A:ED.COM
55	55	12k	1	R-O (A:PIP.COM)
65536	128	16k	2	R/W A:X.DAT

where the [S parameter causes the Size field to be displayed. Without the \$S, STAT skips the Size field and displays the remaining fields. The Size field lists the virtual file size in records; the Recs field sums the number of virtual records in each extent. For files constructed sequentially, the Size and Recs fields are identical.

The Bytes field lists the number of bytes allocated to a file. The system configuration determines the minimum allocation unit at configuration time. The number of bytes corresponds to the record count plus the remaining unused space in the last allocated block for sequential files. Random access files are given data areas only when written, so the Bytes field contains the only accurate allocation figure. For random access files, the Size field gives the logical end-of-file record position, and the Recs field counts the logical records of each extent. Each of these extents, however, can contain unallocated holes, even though they are added into the record counts.

The Ext field counts the number of physical extents allocated to the file. The Ext count corresponds to the file's number of directory entries. Depending on allocation size, a single directory entry can directly address up to 128 kilobytes (8 logical extents). (A physical extent can address up to 256 kilobytes.)

The Att field gives the Read/Write or Read-Only file indicator; the following commands can change the indicator. Similarly, the parentheses enclosing the PIP.COM filename indicate that the file's system indicator is set, so it is not listed in response to DIR commands.

Set file indicators

Syntax: STAT x:filename.typ [R-O
STAT x:filename.typ [R/W
STAT x:filename.typ [SYS
STAT x:filename.typ [DIR

Explanation:

The 4 preceding command forms set or reset permanent file indicators. The R-O indicator places the file (or set of files) in a Read-Only status. A subsequent STAT command can change this status. The R-O status is recorded in the directory with the file, so the file remains R-O through intervening cold start operations. The R/W indicator places the file in permanent Read/Write status. The SYS indicator attaches the system indicator to the file. The DIR command removes the system indicator. The filename and filetype can contain wildcard characters, but files whose attributes are changed are listed at the console when the change occurs. The drive specifier denoted D: is optional.

Attempts to erase or write into a file marked R-O result in the BDOS message

CP/M Error on x: Read-Only File

The BDOS waits for console input before performing a subsequent warm start. (A RETURN is sufficient.)

Display drive characteristics

Syntax: STAT DSK:
STAT x:DSK:

Explanation:

The second command form displays the characteristics of drive X (A through P). STAT lists the drive characteristics in the following form:

```

x: Drive Characteristics
65536: 128-Byte Record Capacity
8192: Kilobyte Drive Capacity
128: 32-Byte Directory Entries
0: Checked Directory Entries
1024: Records/Extent
128: Records/Block
58: Sectors/Track
2: Reserved Tracks

```

where X is the drive specifier (A through P), followed by total record capacity (65536 for an 8-megabyte drive), followed by the total capacity listed in kilobytes. The directory size is listed next, followed by the checked entries. The number of checked entries is, usually, identical to the directory size for removable media, because this mechanism detects changed media during Personal CP/M operation without an intervening warm start. For fixed media, the number is usually zero, because the media are not changed without at least a cold or warm start.

The number of records per extent determines the addressing capacity of each directory entry (1024 times 128 bytes, or 128 kilobytes in the previous example). The number of records per block shows the basic allocation size (in the example: 128 records/block times 128 bytes per record = 16 kilobytes per block).

The number of physical sectors per track, and the number of reserved tracks, follows the listing. For logical drives that share the same physical disk, the number of reserved tracks can be quite large, because this mechanism skips lower-numbered disk areas allocated to other logical disks.

If you use the first command form, STAT produces a drive characteristics table for all currently active drives.

Display users

Syntax: STAT USR:

Explanation:

This command produces a list of user numbers that have files on the currently-addressed disk. The display format is

```

Active User: 0
Active Files: 0 1 3

```

where the first line lists the currently-addressed user number, as set by the last USER command, followed by a list of user numbers scanned from the current directory. In this example, the active user number is 0 (default at cold start), with three user numbers that have active files on the current disk. The operator can subsequently examine the directories of the other user numbers, by logging in with USER 1 or USER 3 commands, followed by a DIR command at the system prompt of Personal CP/M.

SUBMIT Command

Syntax: SUBMIT filename.SUB parm#1 ... parm#N

Explanation:

The SUBMIT command allows you to batch Personal CP/M commands for automatic processing. The SUBMIT command must use the filename of an existing file on the currently-logged disk with the filetype SUB. The SUB file contains Personal CP/M commands, with possible parameter substitution. SUBMIT substitutes the actual parameters PARM#1 ... PARM#N into the Personal CP/M commands and, if no errors occur, Personal CP/M processes the file of substituted commands sequentially.

You can create a Personal CP/M SUB file with the ED program, interspersing "\$" parameters of the form

```
$1 $2 $3 ... $n
```

corresponding to the number of actual parameters to be included when you submit the file for execution. When the SUBMIT transient program executes, the actual parameters PARM#1 ... PARM#N are paired with the formal parameters

\$1 ... \$n in the Personal CP/M commands. If the number of formal and actual parameters do not correspond, the SUBMIT transient program aborts with an error message at the console. The SUBMIT transient program creates a file of substituted commands on the logged disk with the name \$\$\$SUB.

Personal CP/M warm starts when SUBMIT terminates. Then, the CCP reads this command file as a source of input, rather than the console. If you perform the SUBMIT command on any disk other than drive A, the commands are not processed until you insert the disk into drive A and reboot the system. The SUBMIT transient program can access a SUB file on an alternate drive when you precede the filename with a drive specifier. Because SUB files are acted upon only when they appear on drive A, you can create a SUB file on drive B to execute at a later time, when inserted in drive A.

You can abort command processing at any time by pressing the DEL key when the command is read and echoed. In this case, the \$\$\$SUB file is removed and subsequent commands come from the console. Command processing also aborts when the CCP detects an error in any command. Programs that execute under Personal CP/M abort processing command files when error condition occur by erasing any existing \$\$\$SUB file.

To introduce dollar signs into a SUB file, type a \$\$, which reduces to a single \$ within the command file. A ^ character preceding an alphabetic character x produces a single Ctrl-X character within the file.

The last command in a SUB file can initiate another SUB file, allowing chained batch commands.

The utility program called XSUB extends the power of the SUBMIT facility to include line input to programs, as well as to the CCP. The XSUB command must be the first line of the SUB file. When it executes, XSUB self-relocates directly below the CCP. XSUB processes all subsequent SUB command lines, so that programs that read buffered console input (BDOS Function 10) receive their input directly from the SUB file.

The XSUB program remains in memory, and prints the message

```
(XSUB active)
```

on each warm start operation, to indicate its presence. Subsequent submit command streams do not require the XSUB, unless a cold start has intervened.

Examples:

Suppose the file ASMBL.SUB exists on disk, and contains the following commands:

```
ASM $1
DIR $1.*
ERA *.BAK
PIP $2:=$1.PRN
ERA $1.PRN
```

and you issue the command:

```
A>submit asmb1 x prn
```

The SUBMIT transient program reads the ASMBL.SUB file, substituting X for all occurrence of \$1, and PRN for all of occurrence of \$2. This results in a \$\$\$SUB file containing the commands:

```
ASM X
DIR X.*
ERA *.BAK
PIP PRN:=X.PRN
ERA X.PRN
```

The CCP then executes these commands in sequence.

XSUB example: The file SAVER.SUB contains the lines:

```
XSUB
SID
I$1.COM
R
GO
SAVE 1 $2.COM
```

with a subsequent command:

```
A>submit saver pip y
```

SUBMIT substitutes PIP for \$1, and Y for \$2 in the command stream. The XSUB program loads, followed by SID, to which is sent the command lines IPIP.COM, R, and G0, thus returning to the CCP. The CCP then processes the final command SAVE 1 Y.COM, saving only the first "page" of PIP.

TYPE Command

Syntax: TYPE {x:}filename.typ

Explanation:

The drive specifier X (A through P) is optional, and filename and filetype must contain no wildcard characters.

This command displays the contents of the ASCII file FILENAME.TYP at the console device. FILENAME.TYP must be on drive X if the drive is specified, or on the currently-logged drive if no drive is specified.

The TYPE command expands ASCII tabs (Ctrl-I, 09H) characters, assuming that tab positions are set at every eight column.

Examples:

A>>**type b:x.prn**

Displays the file X.PRN on drive B.

USER Command

Syntax: USER n

Explanation:

n is an integer value in the range 0 to 15.

The USER command allows you to maintain separate groups of files (that is to say: separate user areas) in the same directory. User numbers range from 0 to 15.

On the cold boot, you are automatically logged on as user zero. Issue the USER command to move to another user area within the same directory.

Files that are active when you are logged onto one user area remain active when you log onto another user area.

The active user number is maintained until changed by a subsequent USER command, or until a cold boot when user zero is again assumed.

Examples:

A>>**user 2**

Allows you to access to all files in user area 2.

Section 6: ED, the Personal CP/M Context Editor

ROCHE> I have chosen, with regret, since I used it a lot on my NorthStar Horizon, not to retype the ED section. I think that it will repulse CP/M Newbies, that it is too difficult for them. Since I am trying to resurrect Personal CP/M, I can only counsel them to use WordStar Version 4.0 instead, the most powerful 8-bit word-processor ever made.

In Non-document mode, WS4 can edit source code files as big as the disk on which they are stored. In Document mode, you have a very, very extensive list of features to embellish your text. Many books have been written with WS4, so it is a really powerful tool that was widely used by individuals and companies during CP/M days. You will not need another editor under Personal CP/M.

Section 7: ASM, the Personal CP/M assembler

ROCHE> I have decided, with regret, not to retype the ASM section, which was the standard Intel 8080 absolute assembler for CP/M, since Personal CP/M has been rewritten in Zilog Z-80 mnemonics.

In addition, to generate a Personal CP/M system file, you need to use LINK-80,

which is documented in the "Programmer's Utilities Guide for the CP/M Family of Operating Systems". And this "Programmer's Utilities Guide" contains the documentation of MAC, the Intel 8080 macro assembler for CP/M, which is what 99% of people will need to program applications under Personal CP/M. Historically, ASM was provided so that the first users of CP/M could assemble the BIOS for their own hand-made systems. But MAC is so much more powerful that nobody (even me!) continue to use ASM for serious programming. The debugger which works in conjunction with MAC is SID, which is documented in the "SID User's Guide".

(If you want to re-assemble the BDOS, or do some Z-80 programming, use the standard Z-80 macro assembler under CP/M: M80 Version 3.44, and the Z-80 debugger ZSID Version 2.5.)

Section 8: DDT, the Personal CP/M debugging tool

ROCHE> As explained in Section 7, I have decided not to retype the DDT section, since people still programming application programs under Personal CP/M will, of course, use SID, the improved version of DDT which works in conjunction with MAC. SID is documented in the "SID User's Guide".

Appendix A: Personal CP/M messages

Messages come from several different sources. Personal CP/M displays error messages when there are errors in calls to the Basic Disk Operating System (BDOS). Personal CP/M also displays messages when there are errors in command lines. Each utility supplied with Personal CP/M has its own set of messages. The following table lists Personal CP/M messages and utility messages. You might see messages other than those listed here if you are running an application program. Check the application program's documentation for explanations of those messages.

Table A-1. Personal CP/M error messages

Format: Message
Meaning

ABORTED
PIP. You stopped a PIP operation by pressing a key.

BAD DELIMITER
STAT. Check command line for typing errors.

Bad Load
CCP error message, or SAVE error message.

CP/M Error on x: Disk I/O
The disk I/O error results from an error condition returned to the BDOS from the BIOS module. The BDOS makes BIOS read and write calls to execute file-related BDOS calls. If the BIOS read or write routine detects an error, it returns an error code to the BDOS, resulting in this error message.

CP/M Error on x: Invalid Drive
The invalid drive error results from an error condition returned to the BDOS from the BIOS module. The BDOS makes a BIOS SELECT DISK call prior to accessing a drive to perform a requested BDOS function. If the BIOS does not support the selected disk, the BDOS returns an error code, resulting in this error message.

CP/M Error on x: Read-Only File
The Read-Only file error is returned when a program attempts to write to a file that is marked with the Read-Only attribute. It is also returned to a program that attempts to write to a system file opened under user zero from a non-zero user number.

CP/M Error on x: Read-Only Disk
The Read-Only disk error is returned when a program writes to a disk that is in Read-Only status. A drive can be placed in Read-Only status explicitly with the BDOS WRITE PROTECT DISK function.

CANNOT CLOSE DESTINATION FILE--{filespec}
PIP. An output file cannot be closed. Take appropriate action after checking to see if the correct disk is in the drive, and that the drive is not write protected.

Cannot close, R-O
CANNOT CLOSE FILES

Personal CP/M cannot write to file. This usually occurs because the disk is write protected.

SUBMIT. This error can occur during SUB file processing. Check if the correct system disk is in drive A, and that the disk is not write protected. The submit job can be restarted after rebooting Personal CP/M.

CANNOT READ

PIP. PIP cannot read the specified source. Reader might not be implemented.

CANNOT WRITE

PIP. The destination specified in the PIP command is illegal. You probably specified an input device as a destination.

Checksum error

PIP. An Intel HEX file checksum error was encountered. The hex record that produced the error must be corrected, probably by recreating the HEX file.

CHECKSUM ERROR

LOAD ADDRESS hhhh
ERROR ADDRESS hhhh
BYTES READ:

hhhh:
LOAD. An Intel HEX file contains incorrect data. Regenerate the HEX file from the source.

Command Buffer Overflow

SUBMIT. The SUBMIT buffer allows up to 2048 characters in the input file.

Command too long

SUBMIT. A command in the SUB file cannot exceed 125 characters.

CORRECT ERROR, TYPE RETURN OR CTRL-Z

PIP. A checksum error was encountered during the transfer of an Intel HEX file. Correct the HEX file with the checksum error, probably by recreating the HEX file.

DESTINATION IS R-O, DELETE (Y/N)?

PIP. The destination file specified in a PIP command already exists, and it is Read-Only. If you type Y, the destination file is deleted before the copy is done.

Directory full

SUBMIT. There is not enough directory space to write the \$\$\$SUB file used for processing the SUB file. Erase some files, or select a new disk and retry.

DISK READ ERROR-{filespec}

PIP. The input disk file specified in a PIP command cannot be read properly. This is usually the result of an unexpected end-of-file. Correct the problem in your file.

DISK WRITE ERROR-{filespec}

PIP. A disk write operation cannot be successfully performed during a PIP command, probably due to a full disk. Erase some unnecessary files, or get another disk with more space, and execute PIP again.
SUBMIT. The SUBMIT program cannot write the \$\$\$SUB file to the disk. Erase some files, or select a new disk, and try again.

ERROR: BAD PARAMETER

PIP. You entered an illegal parameter in a PIP command. Retype the entry correctly.

ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS hhhh

LOAD. Displayed if LOAD cannot find the specified file, or if no filename is specified.

ERROR: CANNOT CLOSE FILE, LOAD ADDRESS hhhh

LOAD. Caused by an error code returned by a BDOS function call. Disk might be write protected.

ERROR: DISK READ, LOAD ADDRESS hhhh

LOAD. Caused by an error code returned by a BDOS function call.

ERROR: DISK WRITE, LOAD ADDRESS hhhh

LOAD. Destination disk is full.

ERROR: INVERTED LOAD ADDRESS, LOAD ADDRESS hhhh

LOAD. The address of a record was too far from the address of the previously-processed record. This is an internal limitation of LOAD, but it can be circumvented. Use SID to read the Intel HEX file into memory, then use a SAVE command to store the memory image on disk, in a COM file.

ERROR: NO MORE DIRECTORY SPACE, LOAD ADDRESS hhhh
LOAD. Disk directory is full.

Error on line nnn message
SUBMIT. The SUBMIT program displays its messages in the format shown above,
where nnn is the line number of the SUB file. Refer to the message following
the line number.

FILE EXISTS

You have asked Personal CP/M to create or rename a file, using a specification
that is already assigned to another file. Delete the existing file, or use
another file specification.

REN. The new name specified is the name of a file that already exists. You
cannot rename a file with the name of an existing file. If you want to replace
an existing file with a newer version of the same file, either rename or erase
the existing file, or use the PIP utility.

File Not Found

Personal CP/M cannot find the specified file. Check that you have entered the
correct drive specifier, or that you have the correct disk in the drive.

STAT. STAT cannot find the specified file. The message might appear if you
omit the drive specifier. Check to see if the correct disk is in the drive.

FILE NOT FOUND-{filespec}

PIP. You specified an input file that does not exist.

Invalid Assignment

STAT. You specified an invalid drive or file assignment, or misspelled a
device name. This error message might be followed by a list of the valid file
assignments that can follow a filename. If an invalid drive assignment was
attempted, the message: "Use: x:=R-O" is displayed, showing the proper syntax
for drive assignments.

Invalid control character

SUBMIT. The only valid control characters in the submit files of type SUB are
^A through ^Z. Note that, in a SUB file, the control character is represented
by typing the circumflex, ^, not by pressing the control key ("Ctrl").

INVALID DIGIT-{filespec}

PIP. An invalid hex digit has been encountered while reading an Intel HEX
file. Correct the HEX file with the invalid hex digit, usually by recreating
the HEX file.

Invalid Disk Assignment

STAT. Might appear if you follow the drive specification with anything except
=R-O.

Invalid File Attribute

STAT. Appears if you do not specify R-O, R/W, DIR, or SYS.

INVALID FORMAT

PIP. The format of your PIP command is illegal. See the documentation of PIP.

INVALID HEX DIGIT

LOAD ADDRESS hhhh
ERROR ADDRESS hhhh
BYTES READ:
hhhh

LOAD. File contains incorrect hex digit. Recreate the Intel HEX file from its
source file.

INVALID SEPARATOR

PIP. You have placed an invalid character for a separator between two input
filenames.

INVALID USER NUMBER

PIP. You have specified a user number greater than 15. User numbers are in the
range 0 to 15.

n?

USER. You specified a user number greater than 15. For example, if you type
USER 18, the screen displays "18?".

NO DIRECTORY SPACE-{filespec}

PIP. There is not enough directory space for the output file. Erase some
unnecessary files, or get another disk with more directory space, and execute
PIP again.

NO FILE-{filespec}

DIR, ERA, REN, PIP. Personal CP/M cannot find the specified file, or no files
exist.

NO INPUT FILE PRESENT ON DISK
DUMP. The file that you requested does not exist.

No memory
There is not enough memory available for loading the program specified.

NO SPACE
SAVE. Too many files are already on the disk, or no room is left on the disk to save the information.

No SUB file present
SUBMIT. For SUBMIT to operate properly, you must create a file with filetype of SUB. The SUB file contains usual Personal CP/M commands. Use one command per line.

NOT A CHARACTER SOURCE
PIP. The source specified in your PIP command is illegal. You have probably specified an output device as a source.

** NOT DELETED **
PIP. PIP did not delete the file, which might have had the R-O attribute.

NOT FOUND
PIP. PIP cannot find the specified file.

Parameter error
SUBMIT. Within the submit file of type SUB, valid parameters are \$0 through \$9.

QUIT NOT FOUND
PIP. The string argument to a Q command was not found in your input line.

Read error
TYPE. An error occurred when reading the file specified in the TYPE command. Check the disk, and try again. The STAT filespec command can diagnose trouble.

READER STOPPING
PIP. Reader (= auxiliary input device) operation interrupted.

Record Too Long
PIP. PIP cannot process a record longer than 128 bytes.

Requires CP/M 2.0 or later
XSUB. XSUB requires the facilities of CP/M Version 2.0 or newer.

Requires Personal CP/M 1.0 or newer for operation
PIP. This version of PIP requires the facilities of Personal CP/M Version 1.0 or newer.

START NOT FOUND
PIP. The string argument to an S command was not found in your input line.

"SYSTEM" FILE NOT ACCESSIBLE
You tried to access a file set to SYS with the STAT command.

** TOO MANY FILES **
STAT. There is not enough memory for STAT to sort the files specified, or more than 512 files were specified.

UNEXPECTED END OF HEX FILE-{filespec}
PIP. An end-of-file was encountered prior to a termination hex record. The Intel HEX file without a termination record should be corrected, probably by recreating the HEX file.

Unrecognized Destination
PIP. Check command line for valid destination.

Use: STAT x:=R-O
STAT. An invalid STAT drive command was given. The only valid drive assignment in STAT is "STAT x:=R-O", where X is a drive specifier (A through P).

VERIFY ERROR:-{filespec}
PIP. When copying with the V option, PIP found a difference when re-reading the data just written, and comparing it to the data in its memory buffer. Usually, this indicates a failure of either the destination disk or drive.

WRONG CP/M VERSION (REQUIRES 2.0)
XSUB ACTIVE
SUBMIT. XSUB has been invoked.

XSUB ALREADY PRESENT
SUBMIT. XSUB is already active in memory.

YOUR INPUT?

If Personal CP/M cannot find the command that you specified, it redisplayes the command line that you entered in uppercase, followed by a question mark. Check that you have typed the command line correctly, or that the command that you requested exists as a COM file on the default or specified disk.

Appendix B: ASCII and hexadecimal conversions

ASCII stands for American Standard Code for Information Interchange. The code contains 96 printing and 32 non-printing characters used to store data on a disk. Table B-1 defines ASCII symbols; Table B-2 lists the ASCII and hexadecimal conversions. Table B-2 includes binary, decimal, hexadecimal, and ASCII conversions.

Table B-1. ASCII Symbols

Symbol	Meaning	Symbol	Meaning
ACK	acknowledge	FS	file separator
BEL	bell	GS	group separator
BS	backspace	HT	horizontal tabulation
CAN	cancel	LF	line feed
CR	carriage return	NAK	negative acknowledge
DC	device control	NUL	null
DEL	delete	RS	record separator
DLE	data link escape	SI	shift in
EM	end of medium	SO	shift out
ENQ	enquiry	SOH	start of heading
EOT	end of transmission	SP	space
ESC	escape	STX	start of text
ETB	end of transmission	SUB	substitute
ETX	end of text	SYN	synchronous idle
FF	form-feed	US	unit separator
		VT	vertical tabulation

Table B-2. ASCII Conversion Table

Binary	Decimal	Hexa	ASCII
00000000	0	0	NUL (Ctrl-@)
00000001	1	1	SOH (Ctrl-A)
00000010	2	2	STX (Ctrl-B)
00000011	3	3	ETX (Ctrl-C)
00000100	4	4	EOT (Ctrl-D)
00000101	5	5	ENQ (Ctrl-E)
00000110	6	6	ACK (Ctrl-F)
00000111	7	7	BEL (Ctrl-G)
00001000	8	8	BS (Ctrl-H)
00001001	9	9	HT (Ctrl-I)
00001010	10	A	LF (Ctrl-J)
00001011	11	B	VT (Ctrl-K)
00001100	12	C	FF (Ctrl-L)
00001101	13	D	CR (Ctrl-M)
00001110	14	E	SO (Ctrl-N)
00001111	15	F	SI (Ctrl-O)
00010000	16	10	DLE (Ctrl-P)
00010001	17	11	DC1 (Ctrl-Q)
00010010	18	12	DC2 (Ctrl-R)
00010011	19	13	DC3 (Ctrl-S)
00010100	20	14	DC4 (Ctrl-T)
00010101	21	15	NAK (Ctrl-U)
00010110	22	16	SYN (Ctrl-V)
00010111	23	17	ETB (Ctrl-W)
00011000	24	18	CAN (Ctrl-X)
00011001	25	19	EM (Ctrl-Y)
00011010	26	1A	SUB (Ctrl-Z)
00011011	27	1B	ESC (Ctrl-[)
00011100	28	1C	FS (Ctrl-\)
00011101	29	1D	GS (Ctrl-])
00011110	30	1E	RS (Ctrl-^)
00011111	31	1F	US (Ctrl-_)
00100000	32	20	(SPACE)
00100001	33	21	!
00100010	34	22	"

00100011	35	23	#
00100100	36	24	\$
00100101	37	25	%
00100110	38	26	&
00100111	39	27	'
00101000	40	28	(
00101001	41	29)
00101010	42	2A	*
00101011	43	2B	+
00101100	44	2C	,
00101101	45	2D	-
00101110	46	2E	.
00101111	47	2F	/
00110000	48	30	0
00110001	49	31	1
00110010	50	32	2
00110011	51	33	3
00110100	52	34	4
00110101	53	35	5
00110110	54	36	6
00110111	55	37	7
00111000	56	38	8
00111001	57	39	9
00111010	58	3A	:
00111011	59	3B	;
00111100	60	3C	<
00111101	61	3D	=
00111110	62	3E	>
00111111	63	3F	?
01000000	64	40	@
01000001	65	41	A
01000010	66	42	B
01000011	67	43	C
01000100	68	44	D
01000101	69	45	E
01000110	70	46	F
01000111	71	47	G
01001000	72	48	H
01001001	73	49	I
01001010	74	4A	J
01001011	75	4B	K
01001100	76	4C	L
01001101	77	4D	M
01001110	78	4E	N
01001111	79	4F	O
01010000	80	50	P
01010001	81	51	Q
01010010	82	52	R
01010011	83	53	S
01010100	84	54	T
01010101	85	55	U
01010110	86	56	V
01010111	87	57	W
01011000	88	58	X
01011001	89	59	Y
01011010	90	5A	Z
01011011	91	5B	[
01011100	92	5C	\
01011101	93	5D]
01011110	94	5E	^
01011111	95	5F	_
01100000	96	60	`
01100001	97	61	a
01100010	98	62	b
01100011	99	63	c
01100100	100	64	d
01100101	101	65	e
01100110	102	66	f
01100111	103	67	g
01101000	104	68	h
01101001	105	69	i
01101010	106	6A	j
01101011	107	6B	k
01101100	108	6C	l
01101101	109	6D	m
01101110	110	6E	n
01101111	111	6F	o
01110000	112	70	p
01110001	113	71	q
01110010	114	72	r
01110011	115	73	s

01110100	116	74	t
01110101	117	75	u
01110110	118	76	v
01110111	119	77	w
01111000	120	78	x
01111001	121	79	y
01111010	122	7A	z
01111011	123	7B	{
01111100	124	7C	
01111101	125	7D	}
01111110	126	7E	~
01111111	127	7F	DEL

Appendix C: Filetypes

Personal CP/M identifies every file by a unique file specification, which consists of a drive specifier, a filename, and a filetype. The filetype is an optional 3-character ending, separated from the filename by a period ("."). The filetype indicates a special kind of file. Table C-1 lists common filetypes and their meanings.

Table C-1. Common filetypes

Type	Meaning
ASC	File containing ASCII characters. Can be displayed on console with the TYPE command.
ASM	Assembly language source file; the Personal CP/M assemblers assemble or translate a type ASM file into machine language.
BAK	Backup file created by text editor; the editor renames the source file with this filetype, to indicate that the original file has been processed. The original file stays on disk as the backup file, so you can refer to it.
BAS	BASIC source file.
BIN	File containing binary data (usually not displayable with ASCII characters).
COM	Machine language program providing a COMmand to Personal CP/M.
HEX	Program file in Intel HEX file format.
IRL	Indexed REL file produced by LIB-80.
LIB	Used by MAC and RMAC for macro libraries. Printable file displayable on console or printer.
OVL	Program overlay file. PL/I-80 compiler overlay files; you can create overlay files with LINK-80.
PLI	PL/I-80 source program.
PRN	Printable file displayable on console or printer.
REL	Relocatable file produced by RMAC and PL/I-80, that can be linked by LINK-80.
SUB	Filetype required for submit file containing one or more Personal CP/M commands. The SUBMIT program executes commands in files of type SUB, providing a batch execution mode for Personal CP/M.
SYM	Symbol table file. MAC, RMAC, and LINK-80 output files of type SYM. SID and ZSID read files of type SYM.
SYS	System file for Personal CP/M.
WS4	WordStar Version 4.0 document file.
XRF	Cross-reference file produced by XREF.
\$\$\$	Temporary file

Appendix D: Personal CP/M control character summary

Table D-1. Personal CP/M control characters

Char.	Meaning
Ctrl-C	Warm boot (restarts) the Personal CP/M operating system when typed at the beginning of a line.
Ctrl-E	Forces a physical Carriage Return on console, but does not send the command line to Personal CP/M. 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.
Ctrl-J	Sends the command line to Personal CP/M, 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 Personal CP/M, 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 # sign at the current cursor location, moves the cursor to the next line, and displays any partial command that you typed so far.
Ctrl-U	Discards all the characters in the command line (but leaves them displayed), places a # at the current cursor location, and moves the cursor to the next command line.
Ctrl-X	Discards all the character in the command line (actually, removes them from display), and moves the cursor to the beginning of the current line.
DEL	Deletes the last character typed, and echoes it at the console.

Glossary

ambiguous filename:

Filename that contains either of the Personal CP/M wildcard characters, ? or *, in the filename, the filetype, or both. When you use wildcard characters, you create an ambiguous file specification, and can reference more than one Personal CP/M file. See Section 2 of this guide.

applications program:

Program that solves a specific problem. Typical applications programs are business accounting packages, word processing programs, database manager programs, spreadsheet programs, business graphics programs, and communication programs.

argument:

Symbol indicating a place into which you can substitute a number, letter, or name to give an appropriate meaning to a command line.

ASCII:

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

attribute:

File characteristic that can be set ON or OFF.

backup:

Copy of a disk or file made for safe keeping, or the creation of the backup disk or file.

bit:

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

block:

Area of disk.

bootstrap:

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

buffer:

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

built-in commands:

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

byte:

Unit of memory or disk storage containing 8 bits.

character string:

Any combination of letters, numbers, or special characters on your keyboard.

command:

Elements of a Personal CP/M command line. In general, a Personal CP/M command has 3 parts: the command keyword, the command tail, and a Carriage Return keystroke.

command file:

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

command keyword:

Name that identifies an Personal CP/M command, usually the filename of a file of type COM, or a built-in command. The command keyword precedes the command tail and the Carriage Return in the command line.

command syntax:

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

command tail:

Part of a command that follows the command keyword in the command line. The command tail can include a drive specifier, a filename and/or filetype, and options or parameters, but cannot exceed 128 characters. Some commands do not require a command tail.

concatenate:

Term that describes one of PIP's operations, that combines two or more separate files into one new file in the specified sequence.

console:

Primary input/output device. The console consists of a listing device, such as a screen and a keyboard, through which the user communicates with the operating system or applications program.

control character:

Non-printing character combination that sends a simple command to Personal CP/M. Some control characters perform line-editing functions. To enter a control character, hold down the control key ("Ctrl") on your keyboard, and strike the character key specified. See Appendix D.

cursor:

One-character symbol that can appear anywhere on the console screen. The cursor indicates the position where the next keystroke at the console will have an effect.

data file:

Non-executable collection of similar information, that generally requires a command file to manipulate it.

default:

Currently-selected disk drive and/or user number. Any command that does not specify a disk drive or a user number references the default disk drive and user number. When Personal CP/M is first invoked, the default disk drive is drive A, and the default user number is 0, until changed with the USER command.

delimiter:

Special characters that separate different items in a command line. For example, in Personal CP/M, a colon (":") separates the drive specifier from the filename. A period (".") separates the filename from the filetype. Square brackets ("[" and "]") separate any options from their command or filespec. Commas (",") separate one item in an option list from another. All of the preceding special characters are delimiters.

directory:

A portion of a disk that contains descriptions of each file on the disk. In

response to the DIR command, Personal CP/M displays the filenames stored in the directory.

DIR attribute:

File attribute. A file with the DIR attribute can be displayed by a DIR command. The file can be accessed from the default user number only.

disk, diskette:

Magnetic media used to store information. Programs and data are recorded on the disk in the same way that music is recorded on a cassette tape. The term diskette refers to smaller-capacity, removable floppy diskettes. Disk can refer to a diskette, a removable cartridge disk, or a fixed hard-disk.

disk drive:

Peripheral device that reads and writes on hard or floppy disks. Personal CP/M assigns a letter to each drive under its control. For example, Personal CP/M can refer to the drives in a 4-drive system as A, B, C, and D.

drive specifier:

Letter assigned to a drive. A drive specifier is 1-character long, and must be between A through P. A colon (":") separates the drive specifier from the file specification.

editor:

Utility program that creates and modifies text files. An editor can be used for creation of documents, or creation of code for computer programs. The Personal CP/M editor is invoked by typing the command ED next to the system prompt on the console.

executable:

Ready to be run by the computer. Executable code is a series of instructions that can be carried out by the computer. For example, the computer cannot execute names and addresses, but it can execute a program that prints all those names and addresses on mailing labels.

execute a program:

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

FCB:

See "File Control Block".

file:

Collection of characters, instructions, or data stored on a disk. The user can create files on a disk.

File Control Block:

Structure used for accessing files on disk. Contains the drive specifier, filename, filetype, and other information describing a file to be accessed or created on the disk.

filename:

Name assigned to a file. A filename can include a filename of 1 to 8 characters, and a filetype of 0 to 3 characters. A period (".") separates the filename from the filetype.

file specification:

Unique file identifier. A complete Personal CP/M file specification includes a drive specifier (A through P) followed by a colon (":"), a filename of 1 to 8 characters, a period ("."), and a filetype of 0 to 3 characters. For example, d:filename.typ is a complete Personal CP/M file specification.

filetype:

Extension to a filename. A filetype can be from 0 to 3 characters, and must be separated from the filename by a period ("."). A filetype can tell something about the file. Certain programs require that files to be processed have certain filetypes.

floppy disk:

Flexible magnetic disk used to store information. Floppy disks come in 3.5-, 5 1/4- and 8-inch diameters.

hard disk:

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

hardware:

Physical components of a computer.

HEX file:

ASCII-printable representation of a command (machine language) file.

hexadecimal notation:

Notation for the base 16 number system, using the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F to represent the 16 digits. Machine code is often converted to hexadecimal notation because it can be easily represented by ASCII characters, and therefore displayed on the console or printed on paper (See Appendix B).

input:

Data going into the system, usually from an operator typing at the console, or by a program reading from the disk.

interface:

Object that allows 2 independent systems to communicate with each other, as an interface between hardware and software in a microcomputer.

I/O:

Abbreviation for input/output.

keyword:

See "command keyword".

kilobyte:

1024 bytes, denoted as 1K. 32 kilobytes equal 32K. 1024 kilobytes equal one Megabyte, or over one million bytes.

list device:

Device such as a printer, onto which data can be listed or printed.

logical:

Representation of something that might or might not be the same in its actual physical form. For example, a hard disk can occupy one physical drive, and yet you can divide the available storage on it to appear to the user as if it were in several different drives. These apparent drives are the logical drives.

megabyte:

Over one million bytes; 1024 kilobytes. See "byte" and "kilobyte".

microprocessor:

Silicon chip that is the Central Processing Unit (CPU) of the microcomputer.

operating system:

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

option:

One of many parameters that can be part of a command tail. Use options to specify additional conditions for a command's execution.

output:

Data that the system sends to the console or disk.

parameter:

Value in the command tail that provides additional information for the command. Technically, a parameter is a required element of a command.

peripheral devices:

Devices external to the CPU. For example, terminals, printers, and disk drives are common peripheral devices that are not part of the microprocessor, but are used in conjunction with it.

physical:

Actual hardware of a computer. The physical environment varies from computer to computer.

program:

Series of specially-coded instructions that performs specific tasks when executed by a computer.

prompt:

Characters displayed on the screen to help the user decide what the next appropriate action is. A system prompt is a special prompt displayed by the operating system. The system prompt indicates to the user that the operating system is ready to accept input. The Personal CP/M system prompt is an alphabetic character followed by an angle bracket ("A>"). The alphabetic character indicates the default drive. Some applications programs have their own prompts.

Read-Only:

Attribute that can be assigned to a disk file or a disk drive. When assigned to a file, the Read-Only attribute allows you to read from that file, but not change it. When assigned to a drive, the Read-Only attribute allows you to read any file on the disk, but prevents you from adding a new file, erasing or changing a file, renaming a file, or writing on the disk. The STAT command can set a file or a drive to Read-Only. Every file and drive is either Read-Only or Read/Write. The default setting for drives and files is Read/Write, but an error in resetting the disk or changing media automatically sets the drive to Read-Only until the error is corrected. Files and disk drives can be set to either Read-Only or Read/Write.

Read/Write:

Attribute that can be assigned to a disk file or a disk drive. The Read/Write attribute allows you to read from, and write to, a specific Read/Write file, or to any file on a disk that is in a drive set to Read/Write. A file or drive can be set to either Read-Only or Read/Write.

record:

Collection of data. A file consists of one or more records stored on disk. An Personal CP/M record is 128-bytes long.

R-O:

See "Read-Only".

R/W:

See "Read/Write".

sector:

Portion of a disk track. There are a specified number of sectors on each track.

software:

Specially-coded programs that transmit machine readable instructions to the computer, as opposed to hardware, which is the actual physical components of a computer.

source file:

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

string:

See "character string".

syntax:

Format for entering a given command.

SYS attribute:

File attribute. You can give a file the SYS attribute by using the SYS option in the STAT command. A file with the SYS attribute is not displayed in response to a DIR command; you must use STAT *.*. If you give a file with user number 0 the SYS attribute, you can read and execute that file from any user number on the same drive. Use this feature to make your commonly-used programs available under any user number.

system prompt:

Symbol displayed by the operating system, indicating that the system is ready to receive input. See "prompt".

terminal:

See "console".

track:

Concentric rings dividing a disk. There are 77 tracks on a typical 8-inch floppy disk.

Turn-key application:

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

upward compatible:

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

user number:

Number from 0 to 15 assigned to a file when it is created. User numbers can organize files into 16 file groups.

utility tool:

Program that enables the user to perform certain operations, such as copying files, erasing files, and editing files. Utilities are created for the convenience of programmers and users.

wildcard characters:

Special characters that give Personal CP/M a pattern to match when it searches the directory for a file. Personal CP/M recognizes 2 wildcard characters, ? and *. The ? can be substituted for any single character in a filespec, and the * can be substituted for the filename or the filetype, or both. By placing wildcard characters in a filespec, you create an ambiguous filespec, and can quickly reference one or more files.

Index

(To be done by WS4...)

EOF