

## 5.8 The REN Command

Syntax: REN {new-filespec=old-filespec}

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

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

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

CP/M-68K offers you a second method of using the REN command. Type REN then press RETURN. REN presents the following display:

Enter Old Name:

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

Enter New Name:

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

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

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

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

No File

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

File Already Exists

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

No wildcard filenames

Examples:

A>REN NEWASM.BAS=OLDFILE.BAS

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

A>REN

The system prompts for the filespecs:

Enter Old Name:X.PRN

Enter New Name:Y.PRN

A>

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

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

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

A>REN B:NEWLIST=B:OLDLIST

The file OLDLIST changes to NEWLIST on drive B. Since the second drive specifier, B: is implied by the first one, it is unnecessary in this example. The command line above has the same effect as the following:

A>REN B:NEWLIST=OLDLIST

or

A>REN NEWLIST=B:OLDLIST

**5.9 The STAT (Status) Command****Syntax:**

```

STAT
STAT filespec {RO|RW|SYS|DIR|SIZE}
STAT {d:}DSK: |USR:
STAT VAL: | DEV:

```

**Explanation:**

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

Note that the options following filespec can be enclosed in square brackets [], or be preceded by a dollar \$ sign or by no delimiter as shown in the syntax section above.

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

Unlike other CP/M products, you cannot use STAT in CP/M-68K to set a drive to a Read-Only state. Drives are in a Read-Write state by default and become Read-Only only when you remove a disk on which a file is open. The drive reverts to a Read-Write state whenever you exit a program or type a CTRL-C, either of which causes a warm boot. Please see Section 4.4 for a discussion of terminating programs and the term warm boot.

Except for STAT VAL: and STAT DEV:, the STAT commands allow you to use a drive specifier. In CP/M-68K you can specify a drive that is different from the currently logged-in drive. The drive letter that appears in your system prompt identifies the logged-in drive. Execution of a STAT command with a drive specifier that differs from the logged-in drive does not change the logged-in drive.

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

### 5.9.1 Free Space on Disk

Syntax: STAT {d:}

Explanation: STAT with no command tail reports the amount of free storage space that is available on all on-line disks. This form of the STAT command reports free space for only those disks that have been accessed since CP/M-68K was last started or reloaded. You can find the amount of free space on a particular disk by including the drive specifier in the command tail.

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

d: RW, Free Space: nnK

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

Examples: A>STAT

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

A: RW, Free Space: 16K  
B: RW, Free Space: 32K

A>STAT B:

Suppose drive B has 98 Kilobytes of storage that is free for program and data storage. The following message is displayed on your screen:

B: RW, Free Space: 98K

### 5.9.2 Files - Display File Space and Access Mode

Syntax: STAT filespec {SIZE}

Explanation: This form of the STAT command displays the amount of space in kilobytes used by the specified file. It also displays the Access Mode of the file. STAT accepts wildcards in the filename and filetype part of the command tail. When you include a wildcard in



your file specification, the STAT command displays a list of qualifying files from the default or specified drive, with their file characteristics, in alphabetical order.

Note that the S option following the filespec can be enclosed in square brackets [], or be preceded by a dollar \$ sign, or by no delimiter as shown in the syntax line above.

CP/M-68K supports four file Access Modes:

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

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

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

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

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

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

- The first column displays the number of records used by the file, where each record is 128 bytes in length. This value is listed on your screen under the column marked Recs.
- The second column displays the number of kilobytes used by the file, where each kilobyte contains 1,024 bytes. This value is listed under Bytes.
- The third column displays the number of directory entries used by the file. This value appears under the FCBs column. FCB (File Control Block) is another name for a directory entry.

- The Access Modes are displayed under the Attributes column.
- The file specification, consisting of the drive specifier, filename, and filetype of the file appears under Name on your screen.

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

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

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

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

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

Examples:

A>STAT MY\*.\*

This command tells STAT to display the characteristics of all files that begin with the letters MY, with any filetype at all. Assume that the following three files satisfy the file specification. The screen could display the following:

Drive B:				User 0	
Recs	Bytes	FCBs	Attributes	Name	
16	2K	1	Dir RW	B:MYPROG	.A86
8	1K	1	Dir RO	B:MYTEST	.DAT
32	18K	2	Sys RO	B:MYTRAN	.CMD

Total: 21K 4

B: RW, Free Space: 90K

A>STAT MY\*.\* SIZE

This command causes the same action as the previous command, but includes the Size column in the display. Assume that MYFILE.DAT was written using random access from record number 8 through 15, leaving the first 8 records empty. The virtual file size is 16 records, although the file only consumes eight records. The screen appears as follows:

Drive B:					User 0
Size	Recs	Bytes	FCBs	Attributes	Name
16	16	2K	1	Dir RW	B:MYPROG .A86
16	8	1K	1	Dir RO	B:MYTEST .DAT
32	32	18K	2	Sys RO	B:MYTRAN .CMD

Total:            21K        4

B: RW, Free Space: 90K

### 5.9.3 Set File Access Modes (Attributes)

Syntax:            STAT filespec RO |RW |SYS |DIR

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

The four Access Modes, described above, are

RO  
RW  
SYS  
DIR

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

Examples:            A>STAT LETTER.TXT RO

This command sets the Access Mode for the file LETTER.TXT on the default drive to RO. The following message appears on your screen if the file is present:

LETTER.TXT set to RO

The command:

```
B>STAT A:*.68K SYS
```

sets the Access Mode for all files on drive A, with filetype 68K, to SYS. Given that the three command files PIP, ED, and STAT are present on drive A, the following message appears on your screen:

```
PIP.68K set to SYS  
ED.68K set to SYS  
STAT.68K set to SYS
```

#### 5.9.4 Display Disk Status

Syntax: STAT {d:}DSK:

Explanation: This form of the STAT command displays internal information about your disk system for all on-line disks. The information provided by this command is useful for more advanced programming, and is not necessary for your everyday use of CP/M-68K.

Examples: A>STAT DSK:

This STAT command displays information about drive A in the following form. STAT supplies numbers for n.

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

A>STAT B:DSK:

This command produces the information shown in the previous example for drive B.

### 5.9.5 Display User Numbers with Active Files

Syntax: STAT {d:}USR:

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

User numbers are assigned to files that are created under CP/M-68K. Use this form of the STAT command to determine the active user numbers on a disk.

Examples: A>STAT USR:

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

### 5.9.6 Display STAT Commands and Device Names

Syntax: STAT VAL:

Explanation: STAT VAL: displays the general form of the STAT commands. It also displays the possible physical device names that you can assign to each of the four CP/M-68K logical device names.

Examples: The STAT VAL: display is shown below:

A>STAT VAL:

```
Set Attribute: d:filename.typ [ro] [rw] [sys] [dir]
Disk Status   : DSK: d:DSK:
User Status   : USR: d:USR:
Iobyte Assign:
CON: = TTY: CRT: BAT: UC1:
AXI: = TTY: PTR: UR1: UR2:
AXO: = TTY: PTP: UP1: UP2:
LST: = TTY: CRT: LPT: UL1:
A>
```

### 5.9.7 Display and Set Physical to Logical Device Assignments

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

Explanation:   STAT DEV: displays the current assignments for the four CP/M-68K logical device names: CON:, RDR:, PUN:, and LST:. Use the second form of the above STAT command to change these current assignments. The command STAT VAL: displays the possible physical device names that you can assign to each logical device name. Refer to the part of the STAT VAL: display entitled, Iobyte Assign, shown above.

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

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

Examples:       A>STAT CON: = CRT:

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

A>STAT LST: = LPT:

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

## 5.10 The SUBMIT (Batch Processing) Command

**Syntax:** {SUBMIT} {filespec (argument1 ... argument9)}

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

Normally, you enter commands one line at a time. If you must enter the same sequence of commands several times, you might find it easier to batch the commands together using the SUBMIT command. To do this, you create a file and enter your commands in this file. The file is identified by the filename, and must have a filetype of SUB. When you issue the SUBMIT command, SUBMIT reads the file named by the filespec and prepares it for interpretation by CP/M-68K. When the preparation is complete, SUBMIT sends the file to CP/M-68K line by line, as if you were typing each command.

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

```
DIR
DIR *.BAK
ERA *.BAK
PIP LST:=$1.PRN [T$2 $3 $4]
DIRS *.68K
PIP A:*.LTR
DIR B:
```

### 5.10.1 Creating the SUB File

**Explanation:** The SUB file can contain any valid CP/M-68K command or any valid CP/M-68K command with SUBMIT parameters. You can even reference other SUB files in a SUB file. Imbedded files of this kind are treated as chained files, not nested files. That is, any commands following an imbedded SUB file will be ignored; a SUB file called by another SUB file never returns to the caller after successful execution. For that reason, the last line in a SUB file is the only useful place for an imbedded SUB file.

If the imbedded SUB file cannot be executed, CP/M-68K branches around the nonexecutable SUB file and tries to execute those commands appearing below the imbedded SUB file.

All information presented here is proprietary to Digital Research.

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

```
ERA *.BAK ! DIR *.68K ! PIP $1:=$2:$3[V]
```

contains three distinct CP/M-68K commands on one line of a SUBMIT file. The length of the line must not exceed 128 characters.

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

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

```
ERA $1.BAK
DIR $1
PIP A:$1=A:$2.68K
PIP A:FINISH.SUB=A:$0.SUB
```

and you enter the following SUBMIT command:

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

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

```
ERA SAM.BAK
DIR SAM
PIP A:SAM=A:TEX.68K
PIP A:FINISH.SUB=START.SUB
```

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

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

To include an actual dollar sign \$ in your SUB file,



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

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

and you enter the following SUBMIT command,

```
A>SUBMIT SEARCH B
```

then the translated file contains the following:

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

The net effect in this example is that CP/M-68K looks for all files on drive B that have the filetype of \$\$\$.

### 5.10.2 Executing the SUBMIT Command

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

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

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

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

If the first form is used, CP/M-68K first looks in your current user area for the filename you typed with the filetype of SUB. If that file specification is found there, SUBMIT tries to execute the file. If it is not found in your current user area, CP/M-68K then searches user area 0 for filename.SUB. If it is not found in either user area, CP/M-68K displays on your console your filename with a question mark after it.

If the second form is used, CP/M-68K appends the SUB filetype to your filename and searches for that file specification in your current user area, then in user 0, if need be. If that file specification is found, SUBMIT tries to execute the file. If it is not found, CP/M-68K displays on your console your filename with a question mark after it.

If the third form is used, CP/M-68K assumes you wish to invoke the SUBMIT command because you supplied the SUB filetype. If that file specification is found in your current user area (or user 0, if need be), SUBMIT tries to execute the file. If it is not found, CP/M-68K displays on your console your filename with a question mark after it.

If the fourth form is used, the general rules of file searching as set forth in Section 4.2 take precedence as shown in the following three steps:

- 1) CP/M-68K assumes you are trying to invoke a program file in your current user area, but CP/M-68K does not assume you supplied a filetype. Therefore, CP/M-68K first appends a filetype of 68K to your filename and searches your current user area for the file. If filename.68K is found, CP/M-68K executes that file without invoking SUBMIT.
- 2) If filename.68K is not found, CP/M-68K substitutes a blank filetype for the 68K filetype it appended to your filename and searches for this file specification in your current user area. If such a file exists, it is executed and SUBMIT is again bypassed.
- 3) If the file specification with the blank filetype is not found, CP/M-68K substitutes a SUB filetype for the blank filetype and searches for this file specification in your current user area. If filename.SUB is found, CP/M-68K recognizes that you want to execute a SUBMIT file; control passes to SUBMIT and SUBMIT tries to execute your file.
- 4) If CP/M-68K is unable to find any of the above combinations in your current user area, it searches user area 0 first for your filename.68K, then for your filename.<blank>, and finally for your filename.SUB. If none of these three file specifications are found in either the current user area or user area 0, CP/M-68K displays the command keyword followed by a question mark.

If the fifth form is used, CP/M-68K prompts you for your file specification:

Enter Filename:           

The underscore represents your cursor's position. You may then reply with a filename having a blank

filetype or with a filename having a filetype of SUB. If you provide a blank filetype, CP/M-68K appends the SUB filetype for you. If you provide any filetype other than SUB or blank, CP/M-68K rejects it as inappropriate for SUBMIT and asks you to supply a filename with a filetype of SUB or blank.

If you enter only SUBMIT, the system prompts for the rest of the command. You enter the filespec and arguments.

Example:

A>SUBMIT

and the system prompts:

Enter Filename:

Enter the filespec and arguments here, such as

START B TEX

Another example could be:

A>SUBMIT SUBA

## 5.11 The TYPE (Display File) Built-in

Syntax: TYPE {filespec}

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

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

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

Make sure the file specification identifies a file containing character data.

Examples:

A>TYPE MYPROG.A86

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

A>TYPE B:THISFILE

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

TYPE can also be executed without first specifying the file specification in the command line. If you do this, TYPE prompts you for your file specification:

Enter Filename: \_

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

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

NO FILE

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

## 5.12 The USER (Set User Number) Command

Syntax: USER {number}

Explanation: The USER command sets the current user number. The disk directory can be divided into distinct groups according to a user number. User numbers range from 0 through 15.

When CP/M-68K starts, 0 is the current user number. Executable files created under user 0 are accessible from any other user number. Files that you create under any user number other than 0 are not accessible under any other user number, except through the PIP command. See the G option of the PIP Command.

Example: A>USER 3  
3A>

This command changes the current user number to 3.

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

Example: A>USER

Enter User#: \_

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

Enter User#:5  
5A>

The current user number is now 5 on drive A.

End of Section 5

# Section 6

## The CP/M-68K Editor

### 6.1 Introduction to ED

To do almost anything with a computer you need some way to enter data, some way to give the computer the information you want it to process. The programs most commonly used for this task are called editors. They transfer your keystrokes at the keyboard to a disk file. CP/M-68K's editor is named ED. Using ED, you can easily create and alter CP/M-68K text files.

The correct command syntax for invoking the CP/M-68K editor is given in Section 6.2, "Starting ED." After starting ED, you issue commands that transfer text from a disk file to memory for editing. "ED Operation" details this operation and describes the basic text transfer commands that allow you to easily enter and exit the editor.

Section 6.4, "Basic Editing Commands" details the commands that edit a file. Section 6.5, "Combining ED Commands" describes how to combine the basic commands to edit more efficiently. Although you can edit any file with the basic ED commands, ED provides several more commands that perform more complicated editing functions, as described in Section 6.6, "Advanced ED Commands."

During an editing session, ED may return two types of error messages. Section 6.7, "ED Error Messages," lists these messages and provides examples that indicate how to recover from common editing error conditions.

### 6.2 Starting ED

#### Syntax:

```
ED filespec filespec
```

To start ED, enter its name after the CP/M-68K prompt. The command ED must be followed by a file specification, one that contains no wildcard characters, such as:

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

The file specification, MYFILE.TEX in the above example, specifies a file to be edited or created. The file specification can be preceded by a drive specifier but a drive specifier is unnecessary if the file to be edited is on your default drive. Optionally, the file specification can be followed by a drive specifier, as shown in the following example.

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

All Information Presented Here is Proprietary to Digital Research

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

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

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

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

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

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

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

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

```
NEW FILE
: *
```

**Note:** before starting an editing session, use the STAT command to check the amount of free space on your disk. Make sure that the unused portion of your disk is at least as large as the file you are editing or larger if you plan to add characters to the file. When ED finds a disk or directory full, ED has only limited recovery mechanisms. These are explained in Section 6.7, "ED Error Messages."

### 6.3 ED Operation

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

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

All Information Presented Here is Proprietary to Digital Research.

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

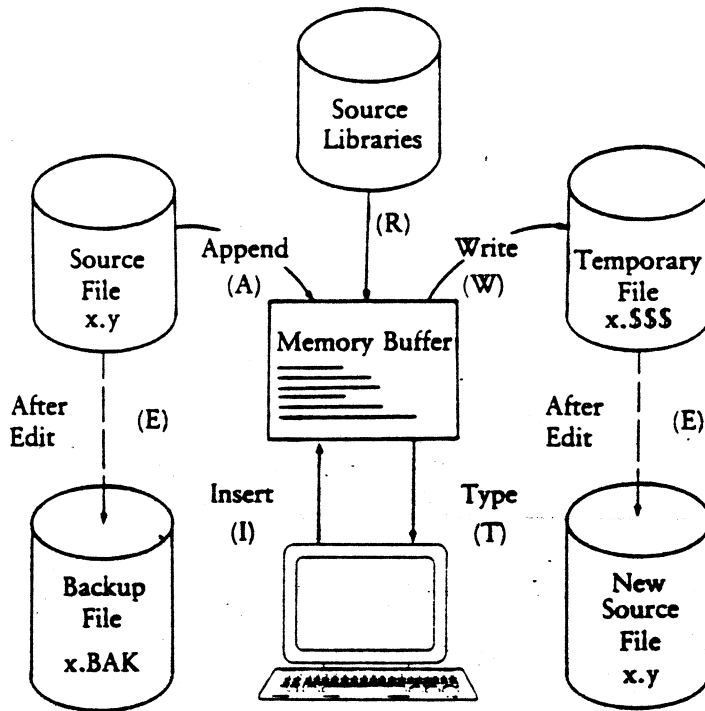


Figure 6-1. Overall ED Operation



In the figure above, the memory buffer is logically between the source file and the temporary work file. ED supports several commands that transfer lines of text between the source file, the memory buffer, and the temporary (and eventually final) file. The following table lists the three basic text transfer commands that allow you to easily enter the editor, write text to the temporary file, and exit the editor.

Table 6-1. Text Transfer Commands

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

### 6.3.1 Appending Text into the Buffer

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

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

#### The A (Append) Command

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

nA

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

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

### 6.3.2 ED Exit

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

#### The W (Write) Command

The W command writes lines from the buffer to the new file. The form of the W command is

```
nW
```

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

```
l: *OV
25000/30000
l: *
```

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

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

#### The E (Exit) Command

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

```
E
```

followed by a carriage return.

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

the .BAK filetype. Finally, ED renames the new file from filename.\$\$\$ to the original filetype and returns control to the CCP.

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

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

#### 6.4 Basic Editing Commands

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

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

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

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

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

Table 6-2. Basic Editing Commands

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

The following sections discuss ED's basic editing commands in more detail. The examples in these sections illustrate how the commands affect the position of the character pointer in the memory buffer. Later examples in Section 6.5, "Combining ED Commands," illustrate how the commands appear at the screen. For these sections, however, the symbol ^ in command examples represents the character pointer, which you must imagine in the memory buffer.

#### 6.4.1 Moving the Character Pointer

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

The B (Beginning/Bottom) Command

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

B, -B

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

The C (Character) Command

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

nC, -nC

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

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

The L (Line) Command

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

nL, -nL

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

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

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

The n (Number) Command

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

n, -n

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

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

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

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

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

**6.4.2 Displaying Memory Buffer Contents**

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

The T (Type) Command

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

nT, -nT

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

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

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

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

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

### 6.4.3 Deleting Characters

#### The D (Delete) Command

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

```
nD, -nD
```

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

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

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

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

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

#### The K (Kill) Command

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

```
nK, -nK
```

where n is the number of lines to be deleted. A positive number

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

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

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

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

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

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

#### 6.4.4 Inserting Characters into the Memory Buffer

##### The I (Insert) Command

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

```
I
Istring^Z
```

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

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

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



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

In command mode, you can use CP/M-68K line editing control characters to edit your input. The table below lists these control characters.

Table 6-3. CP/M-68K Line Editing Controls

Command	Result
CTRL-C	Abort the editor and return to the CP/M-68K system.
CTRL-E	Return carriage for long lines without transmitting command line to the buffer.
CTRL-H	Delete the last character typed on the current line.
CTRL-U	Delete the entire line currently being typed.
CTRL-X	Delete the entire line currently being typed. Same as CTRL-U.
Rubout	Remove the last character and echo deleted character at the screen.

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

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

#### The Istring^Z (Insert String) Command

The second form of the I command does not enter insert mode. It inserts the character string into the memory buffer and returns immediately to the ED prompt. You can use CP/M-68K's line editing control characters to edit the command string.

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

```
iIn 1870, ^Z
```

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

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

### 6.4.5 Replacing Characters

#### The S (Substitute) Command

The S command searches the memory buffer for the specified string, but when it finds it, automatically substitutes a new string for the search string. The S command takes the form:

```
nSsearch string^Znew string
```

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

```
Emily Dickinson^ZThe poet
```

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

```
The poet^ said,<cr><lf>
```

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

### 6.5 Combining ED Commands

It saves keystrokes and editing time to combine the editing and display commands. You can type any number of ED commands on the same line. ED executes the command string only after you press the carriage-return key. Use CP/M-68K's line editing controls to manipulate ED command strings.

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

- The combined-command line must not exceed CP/M-68K's 128 character maximum.
- If the combined-command line contains a character string, the line must not exceed 100 characters.
- Commands to terminate an editing session must not appear in a combined-command line.
- Commands, such as the I, J, R, S, and X commands, that require character strings or filespecs must be either the last command on a line or must be terminated with a CTRL-Z or Esc character, even if no character string or filespec is given.

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

### 6.5.1 Moving the Character Pointer

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

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

```

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

### 6.5.2 Displaying Text

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

line. T types from the CP to the end of the line, allowing you to see the entire line.

```
3: *T
sense of living
3: **OLT
3: the mere sense of living
3: *
```

The command OTT displays the entire line without moving the CP.

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

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

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

### 6.5.3 Editing

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

```
1: *15C5DOLT
1: Emily Dickinson,
1: *
```

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

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

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

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

## 6.6 Advanced ED Commands

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

### 6.6.1 Moving the CP and Displaying Text

#### The P (Page) Command

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

nP, -nP

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

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

#### The n: (Line Number) Command

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

n:

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

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

The :n (Through Line Number) Command

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

```
:ncommand
```

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

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

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

**6.6.2 Finding and Replacing Character Strings**

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

The F (Find) Command

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

```
nFstring
```

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

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

The character pointer moves to the beginning of the third line where the second occurrence of the word "living" is located. To display the line, combine the find command with a type command. Note that if you follow an F command with another ED command on the same line, you must terminate the string with a CTRL-Z, as shown below.

- 1: \*2fliving^Z0lt
- 3: \*the mere sense of living

It makes a difference whether you enter the F command in upper- or lower-case. If you enter F, ED internally translates the argument string to upper-case. If you specify f, ED looks for an exact match. For example, fCp/m-68k searches for CP/M-68K but fCp/m-68k searches for Cp/m-68k and cannot find CP/M-68K, or cp/m-68K.

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

```
BREAK "#" AT
```

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

### The N Command

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

```
nNstring
```

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

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

```
BREAK "#" AT
```

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

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

The J (Juxtapose) Command

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

```
nJsearch string^Zinsert string^Zdelete-to string
```

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

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

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

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

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

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

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

All Information Presented Here is Proprietary to Digital Research



The M (Macro) Command

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

```
nMcommand string
```

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

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

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

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

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

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

**6.6.3 Moving Text Blocks**

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

The X (Transfer) Command

The X command takes the forms:

```
nX
nX filespec^Z
```

where n is the number of lines from the CP towards the bottom of the

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

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

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

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

### The R (Read) Command

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

```
R
Rfilespec
```

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

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

#### 6.6.4 Saving or Abandoning Changes: ED Exit

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

##### The H (Head of File) Command

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

H

followed by a carriage return.

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

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

**A>ED filespec differentfilespec**

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

##### The O (Original) Command

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

O

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

O (Y/N)?

You must respond with either a Y or an N; if you press any other key, ED repeats the question. When you enter Y, ED erases the temporary file and the contents of the memory buffer. When the \*

prompt returns, the character pointer is pointing to the beginning of an empty memory buffer, just as it is when you start ED.

### The Q (Quit) Command

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

Q

followed by a carriage return.

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

Q (Y/N)?

You must respond with either a Y or an N; if you press any other key, ED repeats the question. When you enter Y, ED erases the temporary file, closes the source file, and returns control to CP/M-68K.

**Note:** you can enter a CTRL-C to immediately return control to CP/M-68K. This does not give ED a chance to close the source or new files, but it prevents ED from deleting any temporary files.

## 6.7 ED Error Messages

ED returns one of two types of error messages: an ED error message if ED cannot execute an edit command, or a CP/M-68K error message if ED cannot read or write to the specified file.

The form of an ED error message is

BREAK "x" AT c

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

Table 6-4. ED Error Symbols

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

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

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

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

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

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

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

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

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

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

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

**Table 6-5. ED Disk File Error Messages**

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

End of Section 6

# Appendix A

## Error Messages

Error messages come from several different sources. Each utility supplied with CP/M-68K has its own set of error messages. These messages are displayed when there are errors in the utility command lines. CP/M-68K also displays error messages when there are errors in calls to its built-in system modules: the Basic Disk Operating System (BDOS), the Basic I/O System (BIOS), and the Console Command Processor (CCP). The BDOS provides functions that access the file system, the BIOS provides functions that interface the peripheral device drivers for I/O processing, and the CCP provides the user interface that parses the user command line.

The utility error messages and the CP/M-68K system error messages are listed below in alphabetic order with explanations and suggested user responses. The utility or system module that returns the error message is indicated in the explanation. If you are running an application program, you may see messages other than those listed here. Check the application program's documentation for explanations of those messages.

**Table A-1. CP/M-68K Error Messages**

Message	Meaning
<b>BAD DIRECTORY ON D: SPACE ALLOCATION CONFLICT</b>	<p>STAT error. This message is followed by a list of one or more filenames. The files listed have been doubly allocated and contain data blocks that are already allocated to another file on the disk. The error can be caused by either a hardware or software failure.</p> <p>If you have maintained a backup disk with copies of your files, ERASE the affected files and reboot CP/M-68K. Rebooting the system cleans up the directory by reconstructing the directory allocation vectors. If you do not reboot the system the error will reoccur. To replace the files you erased, COPY or PIP the files from your backup disk.</p>

Table A-1. (continued)

Message	Meaning
	<p>If you do not have a back-up disk, you must recreate the doubly allocated files. If you can tell by careful checking which portions of the files are duplicated or deleted, you can reCOPY or PIP the files from the doubly allocated disk onto a new disk. Use ED to recreate the deleted portions and to delete the parts that were added to the wrong files. To clean up the doubly allocated disk so that it can be used again, ERASE it completely (ERASE *.* ) and reboot CP/M-68K.</p>
	<p>The problem that caused this error may or may not reoccur. If you were running an undebugged program at the time the error occurred, check the program carefully for an error that may have caused the space allocation error. If you were running a program that has been debugged and has caused no problem before, check your hardware. The error may have been caused by a damaged drive or loose connection. If the problem persists, contact your technical support personnel or the place you purchased your system for further assistance.</p>
<b>bad relocation information bits</b>	<p>CCP error. This message is a result of a BDOS Program Load Function 59 error. See the <u>CP/M-68K Operating System Programmer's Guide</u> for more information on Function 59. The message indicates that the file specified in the command line is not a valid executable command file, or that the file has been corrupted.</p> <p>Ensure that the file is a command file. Section 3 of this manual describes the format of a command file. If the file has been corrupted, reassemble or recompile the source file, and relink it before you reenter the command line.</p>



Table A-1. (continued)

Message	Meaning
<b>BIOS ERROR -- DISK X NOT SUPPORTED</b>	<p>BIOS error. The disk drive indicated by the variable X is not supported by the BIOS. The BDOS supports a maximum of 16 drives, lettered A through P. Check the documentation provided by the manufacturer for your system configuration to find out which of the BDOS drives your BIOS implements. Specify the correct drive code and reenter the command line.</p>
<b>BIOS ERROR -- Invalid Disk Status</b>	<p>BIOS error. The disk controller returned unexpected or incomprehensible information to the BIOS. Retry the operation. If the error persists, check the hardware. If the error does not come from the hardware, it is caused by an error in the internal logic of the BIOS. Contact the place you purchased your system for assistance. You should provide the following information:</p> <ul style="list-style-type: none"><li>● Indicate which version of the operating system you are using.</li><li>● Describe your system's hardware configuration.</li><li>● Provide sufficient information to reproduce the error. Indicate which program was running at the time the error occurred. If possible, you should also provide a disk with a copy of the program.</li></ul>

Table A-1. (continued)

Message	Meaning
<b>BREAK "x" AT c</b>	ED error. Where "x" is one of the symbols described below and c is the command letter being executed when the error occurred.
#	Search failure. ED cannot find the string specified in a F, S, or N command.
?	Unrecognized command letter c. ED does not recognize the indicated command letter, or an E, H, Q, or O command is not alone on its command line.
O	The file specified in a R command could not be found.
>	Buffer full. ED cannot put any more characters in the memory buffer, or the string specified in an F, N, or S command is too long.
E	Command aborted. A keystroke at the console aborted command execution.
F	Disk or directory full. This error is followed by either the disk or directory full message. Refer to the recovery procedures listed under these messages.
<b>CANNOT EDIT WILDCARD FILENAME</b>	
	ED error. A wildcard (*) filename was requested when invoking ED. ED does not support wildcard filenames as sources. Request a specific filename and reenter the ED command line.

Table A-1. (continued)

Message	Meaning
<b>CP/M Disk change error on drive x</b>	<p>Disk error. The disk in the drive indicated by the variable x is not the same disk the system logged in previously. When the disk was replaced, you did not enter a CTRL-C to log in the current disk. Therefore, when you attempted to write to, erase, or rename a file on the current disk, the BDOS set the drive status to Read-Only and warm booted the system. The current disk in the drive was not overwritten. The drive status was returned to Read-Write when the system was warm booted.</p> <p>Each time a disk is changed, you must type a CTRL-C to log in the new disk.</p>
<b>CP/M Disk file error: filename is read-only. Do you want to: Change it to read/write (C), or Abort (A)?</b>	<p>Disk error. You attempted to write to, erase, or rename a file whose status is Read-Only. Specify one of the options enclosed in parentheses. If you specify the C option, the BDOS changes the status of the file to Read-Write and continues the operation. The Read-Only protection previously assigned to the file is lost. If you specify the A option or a CTRL-C, the program terminates and CPM-68K returns the system prompt.</p>
<b>CP/M Disk read error on drive x Do you want to: Abort (A), Retry (R), or Continue with bad data (C)?</b>	<p>BDOS error. This message indicates a hardware error. Specify one of the options enclosed in parentheses. Each option is described below.</p> <p>A or CTRL-C Terminates the operation and CP/M-68K returns the system prompt.</p> <p>R Retries the operation. If the retry fails, the system reprompts with the option message.</p>

Table A-1. (continued)

Message	Meaning
C	<p> Ignores the error and continues program execution. Be careful if you use this option. Program execution should not be continued for some types of programs. For example, if you are updating a data base and receive this error but continue program execution, you can corrupt the index fields and the entire data base. For other programs, continuing program execution is recommended. For example, when you transfer a long text file and receive an error because one sector is bad, you can continue transferring the file. After the file is transferred, review the file, and add the data that was not transferred due to the bad sector.</p>
<p><b>CP/M Disk write error on drive x Do you want to: Abort (A), Retry (R), or Continue with bad data (C)?</b></p>	
<p>BDOS error. This message indicates a hardware error. Specify one of the options enclosed in parentheses. Each option is described below.</p>	
A or CTRL-C	<p> Terminates the operation and CP/M-68K returns the system prompt.</p>
R	<p> Retries the operation. If the retry fails, the system reprompts with the option message.</p>
C	<p> Ignores the error and continues program execution. Be careful if you use this option. Program execution should not be continued for some types of programs. For example, if you are updating a data base and receive this error but continue program execution, you can corrupt the index fields and the entire data base. For other programs, continuing program execution is recommended. For example, when you transfer a long text file and receive an error because one sector is bad, you can continue transferring the file. After the file is transferred, review the file, and add the data that was not transferred due to the bad sector.</p>

Table A-1. (continued)

Message	Meaning
CP/M Disk select error on drive x Do you want to: Abort (A), Retry (R)	Disk error. There is no disk in the drive or the disk is not inserted correctly. Ensure that the disk is securely inserted in the drive. If you enter the R option, the system retries the operation. If you enter the A option or CTRL-C the program terminates and CPM-68K returns the system prompt.
CP/M Disk select error on drive x	Disk error. The disk selected in the command line is outside the range A through P. CP/M-68K can support up to 16 drives, lettered A through P. Check the documentation provided by the manufacturer to find out which drives your particular system configuration supports. Specify the correct drive code and reenter the command line.
<b>DIRECTORY FULL</b>	<p>ED error. There is not enough directory space for the file being written.</p> <p>You can use the OXfilespec command to erase any unnecessary files on the disk without leaving the editor. Alternatively, you can save the contents of the memory buffer on another disk with the command, B#Xfilespec, where filespec is a file on a different drive. You can then quit the edit. If you reedit the file, the output should be placed on a different drive with the command, ED filespec d:, where d: is a valid drive name other than the drive containing the source file.</p> <p>You can read the file saved with the Rfilespec command. Caution, part of the file may not be in the memory buffer when you save it (if you have not appended the whole file or if you have issued any W commands).</p>

Table A-1. (continued)

Message	Meaning
<b>DISK FULL</b>	ED error. There is not enough disk space for the output file. This error can occur on the W, E, H, or X commands. If it occurs with the X command you can repeat the command, prefixing the filename with a different drive. Otherwise you can try the recovery methods described above under the Directory Full error.
<b>Disk select error on Destination</b> <b>Disk select error on Source</b> <b>ERROR</b>	BDOS, COPY, INIT error. The disk selected in the command line is not supported by CP/M-68K. CP/M-68K can support up to 16 drives, lettered A through P. Check the documentation provided by the manufacturer to find out which drives your particular system configuration supports. Specify the correct drive code and reenter the command line.
<b>ERROR: BAD PARAMETER</b>	PIP error. The command contains an illegal parameter. Refer to the section in this User's Guide on the PIP command for an explanation and listing of the PIP command parameters. Reenter the command with a valid parameter.

Table A-1. (continued)

Message	Meaning
<b>ERROR: CLOSE FILE - {filespec}</b>	<p>PIP error. The output file specified in the previous PIP command cannot be closed. This message indicates a fatal error in the internal logic of PIP. Contact the place you purchased your system for assistance. You should provide the information below.</p> <ul style="list-style-type: none"> <li>● Indicate which version of the operating system you are using.</li> <li>● Describe your system's hardware configuration.</li> <li>● Provide sufficient information to reproduce the error. Indicate which program was running at the time the error occurred. If possible, you should also provide a disk with a copy of the program.</li> </ul>
<b>ERROR: DESTINATION IS R/O, DELETE (Y/N)?</b>	<p>PIP error. The destination file specified in a PIP command already exists and it is Read-Only. If you type Y, the existing file is deleted and replaced with the file specified in the PIP command. If you do not want to delete the existing file, type N and reenter the PIP command with a different destination file specification: either specify a new filename, or direct the destination file to a different disk. This message can be avoided by using the W option. Refer to the section of this User's Guide on the PIP command for a discussion and list of the PIP options.</p>
<b>ERROR: DISK READ - {filespec}</b>	<p>PIP error. PIP can not read the input file indicated by the variable, filespec. This message indicates a bad disk. Ensure that the disk is in place and retry the operation. If the error persists, the disk is bad and must be replaced. COPY or PIP as much of the bad disk as possible, or use your back-up disk, if you maintained one, to replace the lost files.</p>

Table A-1. (continued)

Message	Meaning
<b>ERROR: DISK WRITE - {filespec}</b>	<p>PIP error. The disk to which PIP is writing is full. The file indicated by the variable, filespec, was not copied. Reenter the PIP command line and redirect the output to another disk, if possible. If you do not have another disk available, you must erase unnecessary files, if any, or replace the disk before you reenter the PIP command line.</p>
<b>ERROR: FILE NOT FOUND - {filespec}</b>	<p>PIP error. The input file indicated by the variable, filespec, does not exist, or the drive code or user number is incorrect. Ensure that you are typing the correct filename and drive code, and that you are in the correct user number or have specified the correct user number in the option brackets before you reenter the PIP command line.</p>
<b>ERROR: HEX RECORD CHECKSUM - {filespec}</b>	<p>PIP error. A hex record checksum error was encountered during the transfer of a hex file. This message indicates a corrupted or truncated hex file. ReCOPY or PIP the hex file from your back-up disks, if possible. If you do not have a back-up, you must ERASE the bad file and recreate it.</p>
<b>ERROR: INVALID DESTINATION</b>	<p>PIP error. The destination specified in your PIP command is illegal. You have specified an input device as a destination, or used a wildcard file specification. Respecify the PIP command with an output device or a specific file as the destination. Refer to the section of this User's Guide on the PIP command for a discussion of the valid PIP destinations.</p>



Table A-1. (continued)

Message	Meaning
<b>ERROR: INVALID FORMAT</b>	PIP error. The format of your PIP command is illegal. Refer to the section of this User's Guide on the PIP command for the correct syntax.
<b>ERROR: INVALID HEX DIGIT - {filespec}</b>	PIP error. An invalid hex digit has been encountered while reading a hex file. This message indicates a truncated or corrupted hex file. ReCOPY or PIP the file from your backup disk. If you do not have a backup, you must ERASE the bad file and recreate it.
<b>ERROR: INVALID SEPARATOR</b>	PIP error. You have placed an invalid character for a separator between two input filenames. Reenter the PIP command line using a comma as the separator between the input filenames.
<b>ERROR: INVALID SOURCE</b>	PIP error. The source specified in your PIP command is illegal. You have specified an output device as a source or used a wildcard filename. Reenter the PIP command line with an input device or a specific filename as the source. Refer to the section of this User's Guide on PIP for a discussion of the legal sources for a PIP command.
<b>ERROR: INVALID USER NUMBER</b>	PIP error. You have specified a user number greater than 15. User numbers are in the range 0 to 15. Reenter the PIP command line using a valid user number.

Table A-1. (continued)

Message	Meaning
<b>ERROR: NO DIRECTORY SPACE - {filespec}</b>	<p>PIP error. The disk to which PIP is writing has no directory space available.</p> <p>ERase unnecessary files, if any, or replace the disk before you reenter the PIP command line.</p>
<b>ERROR: QUIT NOT FOUND</b>	<p>PIP error. The string argument to a Q parameter was not found in your input file. Ensure that you typed the correct string and retry the operation. This error can also occur when entering the Q option from the CP/M-68K command line. CP/M-68K translates everything in the command line into upper-case. To avoid this problem, enter the Q option from inside PIP. Enter PIP, and wait for the * prompt before entering the file specifications and Q option.</p>
<b>ERROR: START NOT FOUND</b>	<p>PIP error. The string argument to an S parameter could not be found in the source file. Ensure that you are typing the correct string and retry the operation. This error can also occur when entering the S option from the CP/M-68K command line. CP/M-68K translates everything in the command line into upper-case. To avoid this problem, enter the S option from inside PIP. Enter PIP, and wait for the * prompt before entering the file specifications and S option.</p>
<b>ERROR: UNEXPECTED END OF HEX FILE - {filespec}</b>	<p>PIP error. The last hex record was not as long as the character count indicated. The hex file is truncated. ReCOPY or PIP the truncated file from your back-up disk. If you do not have a back-up, you must ERase the bad file and recreate it.</p>

Table A-1. (continued)

Message	Meaning
<b>ERROR: USER ABORTED</b>	PIP error. You aborted a PIP operation by entering a CTRL-C or by pressing a key. If you did not intend to abort the operation, you must reenter the PIP command line.
<b>ERROR: VERIFY - {filespec}</b>	PIP error. When copying with the V option, PIP found a difference when rereading the data just written and comparing it to the data in its memory buffer. This indicates a bad disk or a hardware error. Retry the operation. If the error persists, insert a new disk and reenter the PIP command line. If the error still persists, check the hardware.
<b>Extraneous argument ignored argument?</b>	COPY error. The command line contains too many arguments. The extraneous arguments are indicated by the variable ,arguments. The command was carried out up to, but not including, the extraneous arguments. Refer to the section in this User's Guide on COPY for the correct syntax of the COPY command line. Reenter the command line with the correct syntax for the remaining arguments.
<b>File already exists</b>	REN error. The name specified in the command line as the new filename already exists. Use the ERA command to delete the existing file if you wish to replace it with the new file. If not, select another filename and reenter the REN command line.

Table A-1. (continued)

Message	Meaning
<b>FILE IS READ/ONLY</b>	ED error. A Read-Only file cannot be edited with the ED command: ED filespec. The command, ED inputfilespec outputfilespec, should be used instead.
<b>FILE NOT FOUND</b> <b>FILE NOT FOUND - {filespec}</b>	ED, STAT error. ED or STAT could not find the file indicated by the variable, filespec. Check the filename, drive code and user number before you reenter the command line.
<b>FILENAME REQUIRED</b>	ED error. The ED command was typed without a filename. Reenter the ED command followed by the name of the file you wish to edit or create.
<b>Formatting Error</b>	FORMAT error. This message indicates either a bad disk or a hardware error. Retry the operation. If the error reoccurs, replace the disk with a new disk. If the error persists, look for a hardware error. Contact your technical support personnel or the place you purchased your system for assistance.
<b>insufficient memory or bad file header</b>	CCP error. This error could result from one of three causes: <ul style="list-style-type: none"> <li>● The file is not a valid executable command file.</li> <li>● The program is too large for the available memory.</li> <li>● The program is linked to an absolute location in memory that cannot be used.</li> </ul>

Table A-1. (continued)

Message	Meaning
	<p>Ensure that you are requesting the correct file. This error can occur when you enter the filename before you enter the command for a utility. Check the appropriate section of this User's Guide or the <u>CP/M-68K Programmer's Guide</u> for the correct command syntax before you reenter the command line. If you are trying to run a program when this error occurs, the program file may have been corrupted. Reassemble or recompile the source file and relink it before you reenter the command line.</p> <p>Add more memory boards to the system configuration, or rewrite the program to use less memory.</p> <p>The program must be made relocatable, or linked to a usable memory location. The BDOS Get/Set TPA Limits Function (63) returns the high and low boundaries of the memory space that is available for loading programs. See the <u>CP/M-68K Programmer's Guide</u> for more information on Function 63.</p>
<p><b>INVALID ASSIGNMENT</b>  <b>USE: STAT D:FILENAME.TYP [SIZE] [RO] [RW] [SYS] OR [DIR]</b></p>	<p>STAT error. An invalid file assignment was attempted. The error message is followed by the correct syntax and a list of the valid file assignments. Refer to the section of this User's Guide on the STAT command for a discussion of file status assignments. Specify a valid file assignment and reenter the STAT command line.</p>
<p><b>INVALID FILENAME</b></p>	<p>ED error. The filename specified is too long, or a delimiter was used as the first character. Filenames cannot be longer than eight characters. Refer to the section of this User's Guide on naming files for a list and description of the delimiters that cannot be used as the first character in a filename. Correct the filename and reenter the ED command line.</p>

Table A-1. (continued)

Message	Meaning
<b>Invalid or missing drive code</b>	<p>FORMAT error. The FORMAT command line is either missing a drive code specification, or the drive specified is not supported by the BDOS. CP/M-68K can support up to 16 drives, lettered A through P. Check the documentation provided by the manufacturer to find out which drives your particular system configuration supports. Specify the correct drive code and reenter the command line.</p>
<b>No file</b>	<p>CCP, COPY, ERA, REN error. The filename specified in the command line does not exist. Ensure that you use the correct filename and reenter the command line.</p>
<b>NO MEMORY</b>	<p>ED error. There is not enough space in available memory to create the buffer that ED requires to edit a file. Contact your technical support personnel or place of purchase for assistance. This problem can occur when ED is absolutely linked too high in the TPA to allow for the buffer. ED needs 4K of memory at a location immediately higher in memory than itself for a buffer. Relink ED to a lower location. If ED is not linked too high, reconfigure the TPA of your system, if possible, by eliminating some system extensions. If it is not possible to reconfigure the TPA, additional memory boards must be added to the system configuration to run ED.</p>
<b>No wildcard filenames</b>	<p>CCP, REN error. The command specified in the command line does not accept wildcards in file specifications. Response: Retype the command line using a specific filename.</p>

Table A-1. (continued)

Message	Meaning
<b>OUTPUT FILE EXISTS, ERASE IT</b>	<p>ED error. This error occurs when placing the destination file on a different disk than the source. The destination filename already exists on the receiving disk. The existing file should be erased or another disk selected to receive the output file.</p>
<b>Program Load Error</b>	<p>CCP error. This message indicates an undefined failure of the BDOS Program Load Function. See the <u>CP/M-68K Programmer's Guide</u> for more information on Function 59. Reboot the system and try again. If the error persists, then it is caused by an error in the internal logic of the BDOS. Contact the place you purchased your system for assistance. You should provide the following information:</p> <ul style="list-style-type: none"><li>● Indicate which version of the operating system you are using.</li><li>● Describe your system's hardware configuration.</li><li>● Provide sufficient information to reproduce the error. Indicate which program was running at the time the error occurred. If possible, you should also provide a disk with a copy of the program.</li></ul>
<b>READ error ----&gt; Track: t Sector: s</b>	<p>COPY error. This message indicates a bad disk. The variable t is replaced with the number of the track where the error occurred. The variable s is replaced with the number of the bad sector. Retry the operation. If the error persists, use a new disk and COPY or PIP as many files as possible from the damaged disk. If you cannot COPY or PIP a file, you must replace the file from your backup disk. If you have not maintained a backup disk, you must recreate the lost file. Discard the damaged disk.</p>

Table A-1. (continued)

Message	Meaning
<b>read error on program load</b>	<p>CCP error. This message indicates a premature end-of-file. The file is smaller than the header information indicates. Either the file header has been corrupted or the file was only partially written. If the program is one you have written, you must reassemble or recompile the source file, and relink it before you reenter the command line. If this message occurs any other time, and you have a backup copy of the program you were running, erase the program file and replace it from your backup files. If the error reoccurs, contact your technical support personnel or the place you purchased your system for assistance in locating the cause of the error.</p>
<b>READ ONLY DRIVE STATUS NOT ALLOWED</b>	<p>STAT error. Read-Only drive status is not supported by CP/M-68K. Assign each file on the disk Read-Only status individually.</p>
<b>Select Error</b>	<p>FORMAT, INIT error. This message indicates one of three errors:</p> <ul style="list-style-type: none"><li>● The drive specified is not supported by your system.</li><li>● The disk on the drive selected is not in place.</li><li>● A hardware error.</li></ul>



Table A-1. (continued)

Message	Meaning
	<p>CP/M-68K supports a maximum of 16 drives, lettered A through P. Check the documentation provided by the manufacturer to find out which of these drives your particular system configuration supports. Reenter the command line with a valid drive code specification.</p> <p>Check the disk. Ensure that it is securely in place and that the drive door is closed.</p> <p>If the disk is in place and you have selected a valid drive code, look for a hardware error. The error may have been caused by a loose connection or damaged drive. Contact your technical support personnel or the place you purchased your system for further assistance.</p>
<b>Source and Destination must be different</b>	<p>COPY error. The source and destination drives specified are the same. Reenter the COPY command line with a different drive for the source than for the destination.</p>
<b>SUB file not found</b>	<p>SUBMIT error. The file requested either does not exist, or does not have a filetype of SUB. Ensure that you are requesting the correct file. Refer to the section on the SUBMIT command in this User's Guide for information on creating and using submit files.</p>
<b>Syntax: REN newfile=oldfile</b>	<p>REN error. The syntax of the REN command line is incorrect. The correct syntax is given in the error message. Enter the REN command followed by a space, then the new filename, followed immediately by an equals sign and the name of the file you want to rename.</p>

Table A-1. (continued)

Message	Meaning
<b>Too many arguments: argument?</b>	<p>DIR, ERA, TYPE, USER error. The command line contains too many arguments. The extraneous arguments are indicated by the variable, argument. Refer to the appropriate section in this User's Guide for the correct syntax for the command. Specify only as many arguments as the command syntax allows and reenter the command line. Use a second command line for the remaining arguments if appropriate. The command may be repeated on one command line when separated by an exclamation point; for example, DIR a:!DIR b:.</p>
<b>TOO MANY FILES TOO MANY ENTRIES IN INDEX TABLE</b>	<p>STAT error. There is not enough memory for STAT to sort the files specified or more than 512 files were specified. Move some files to another user number, or otherwise break the group of files into smaller groups before you reenter the STAT command line. For example, use a wildcard (*.typ) to run STAT on only those files with a certain filetype.</p>
<b>Undefined option</b>	<p>COPY error. The option specified in the COPY command line is unsupported. The only COPY options supported by CP/M-68K are a, for automatic, and v, for verify. Correct and reenter the COPY command line.</p>
<b>User # range is [0-15]</b>	<p>CCP error. The user number specified in the command line is not supported by the BIOS. The valid range is enclosed in the square brackets in the error message. Specify a user number between 0 and 15 (decimal) when you reenter the command line.</p>

Table A-1. (continued)

Message	Meaning
<b>Write Error</b>	<p>FORMAT, INIT error. This message indicates either a bad disk or a hardware error. Retry the operation. If the error reoccurs, replace the disk with a new disk. If the error persists, look for a hardware error. The error may have been caused by a loose connection or damaged drive. Contact your technical support personnel or the place you purchased your system for further assistance.</p>
<b>WRITE error ----&gt; Track: t Sector: s</b>	<p>COPY error. This message indicates a bad disk. The variable t is replaced with the number of the track where the error occurred. The variable s is replaced with the number of the bad sector. Retry the operation. If the error persists, use a new disk and copy as many files as possible from the damaged disk. If you cannot copy a file, you must replace the file from your backup disk. If you have not maintained a backup disk, you must recreate the lost file. Discard the damaged disk.</p>

End of Appendix A

## Appendix B

### ASCII and Hexidecimal 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, then Table B-2 lists the ASCII and hexadecimal conversions. The table 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	Hexadecimal	ASCII
0000000	0	0	NUL
0000001	1	1	SOH (CTRL-A)
0000010	2	2	STX (CTRL-B)
0000011	3	3	ETX (CTRL-C)
0000100	4	4	EOT (CTRL-D)
0000101	5	5	ENQ (CTRL-E)
0000110	6	6	ACK (CTRL-F)
0000111	7	7	BEL (CTRL-G)
0001000	8	8	BS (CTRL-H)
0001001	9	9	HT (CTRL-I)
0001010	10	A	LF (CTRL-J)
0001011	11	B	VT (CTRL-K)
0001100	12	C	FF (CTRL-L)
0001101	13	D	CR (CTRL-M)
0001110	14	E	SO (CTRL-N)
0001111	15	F	SI (CTRL-O)
0010000	16	10	DLE (CTRL-P)
0010001	17	11	DC1 (CTRL-Q)
0010010	18	12	DC2 (CTRL-R)
0010011	19	13	DC3 (CTRL-S)
0010100	20	14	DC4 (CTRL-T)
0010101	21	15	NAK (CTRL-U)
0010110	22	16	SYN (CTRL-V)
0010111	23	17	ETB (CTRL-W)
0011000	24	18	CAN (CTRL-X)
0011001	25	19	EM (CTRL-Y)
0011010	26	1A	SUB (CTRL-Z)
0011011	27	1B	ESC (CTRL-[)
0011100	28	1C	FS (CTRL-\)
0011101	29	1D	GS (CTRL-])
0011110	30	1E	RS (CTRL-^)
0011111	31	1F	US (CTRL-_)
0100000	32	20	(SPACE)
0100001	33	21	!
0100010	34	22	"
0100011	35	23	#
0100100	36	24	\$
0100101	37	25	%
0100110	38	26	&
0100111	39	27	'
0101000	40	28	(
0101001	41	29	)
0101010	42	2A	*
0101011	43	2B	+
0101100	44	2C	,
0101101	45	2D	-
0101110	46	2E	.
0101111	47	2F	/
0110000	48	30	0
0110001	49	31	1
0110010	50	32	2

Table B-2. (continued)

Binary	Decimal	Hexadecimal	ASCII
0110011	51	33	3
0110100	52	34	4
0110101	53	35	5
0110110	54	36	6
0110111	55	37	7
0111000	56	38	8
0111001	57	39	9
0111010	58	3A	:
0111011	59	3B	;
0111100	60	3C	<
0111101	61	3D	=
0111110	62	3E	>
0111111	63	3F	?
1000000	64	40	@
1000001	65	41	A
1000010	66	42	B
1000011	67	43	C
1000100	68	44	D
1000101	69	45	E
1000110	70	46	F
1000111	71	47	G
1001000	72	48	H
1001001	73	49	I
1001010	74	4A	J
1001011	75	4B	K
1001100	76	4C	L
1001101	77	4D	M
1001110	78	4E	N
1001111	79	4F	O
1010000	80	50	P
1010001	81	51	Q
1010010	82	52	R
1010011	83	53	S
1010100	84	54	T
1010101	85	55	U
1010110	86	56	V
1010111	87	57	W
1011000	88	58	X
1011001	89	59	Y
1011010	90	5A	Z
1011011	91	5B	[
1011100	92	5C	
1011101	93	5D	]
1011110	94	5E	^
1011111	95	5F	<
1100000	96	60	'
1100001	97	61	a
1100010	98	62	b
1100011	99	63	c
1100100	100	64	d

All Information Presented Here is Proprietary to Digital Research

Table B-2. (continued)

Binary	Decimal	Hexadecimal	ASCII
1100101	101	65	e
1100110	102	66	f
1100111	103	67	g
1101000	104	68	h
1101001	105	69	i
1101010	106	6A	j
1101011	107	6B	k
1101100	108	6C	l
1101101	109	6D	m
1101110	110	6E	n
1101111	111	6F	o
1110000	112	70	p
1110001	113	71	q
1110010	114	72	r
1110011	115	73	s
1110100	116	74	t
1110101	117	75	u
1110110	118	76	v
1110111	119	77	w
1111000	120	78	x
1111001	121	79	y
1111010	122	7A	z
1111011	123	7B	
1111100	124	7C	
1111101	125	7D	
1111110	126	7E	
1111111	127	7F	DEL

End of Appendix B

## Appendix C

### File Types

CP/M-68K identifies every file by a unique file specification, which consists of a drive specification, a filename, a filetype, and an optional password. The filetype is an optional three-character ending separated from the filename by a period. The filetype generally indicates a special kind of file. The following table lists common filetypes and their meanings.

**Table C-1. Common Filetypes**

Type	Meaning
<b>S</b>	Assembly language source file; the CP/M-68K Assembler assembles or translates a type .S file into machine language.
<b>BAK</b>	Back-up 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 back-up file, so you can refer to it.
<b>C</b>	Program source file written in the C language.
<b>68K</b>	68000 executable file.
<b>H</b>	Header files for C language programs.
<b>HEX</b>	Program file in hexadecimal format.
<b>O</b>	Compiled or assembled object files.
<b>PRN</b>	Printable file displayable on console or printer.
<b>REL</b>	Relocatable file produced by LO68 when the -R option is used. REL files may be converted to .68K files by using the RELOC utility. RELOC is explained in the <u>CP/M-68K Programmer's Guide</u> .



Table C-1. (continued)

Type	Meaning
<b>SUB</b>	Filetype required for submit file containing one or more CP/M-68K commands. The SUBMIT program executes commands in files of type SUB, providing a batch execution mode for CP/M-68K.
<b>SYS</b>	System file for CP/M-68K.
<b>\$\$\$</b>	Temporary file which is inaccessible by user.

End of Appendix C

## Appendix D

### CP/M-68K Control Character Summary

**Table D-1. CP/M-68K Control Characters**

Keystroke	Action
CTRL-C	Terminates executing transient utility program. Also terminates DIR, DIRS, SUBMIT, and TYPE Built-in programs in progress. When entered as first character in a command line or during the execution of the above-mentioned programs, causes a warm start and the redisplay of the system prompt. (Please see Section 4.4 for a discussion of warm start.)
CTRL-E	Forces a physical carriage return but does not send the command line to CP/M-68K. Moves the cursor to the beginning of the following line without erasing your previous input.
CTRL-H	Deletes a character and moves the cursor left one character position.
CTRL-I	Moves the cursor to the next tab stop. Tab stops are automatically set at each eighth column. Has the same effect as the TAB key.
CTRL-J	Moves the cursor to the left of the current line and sends the command line to CP/M-68K. Has the same effect as a carriage return keystroke.
CTRL-M	Moves the cursor to the left of the current line and sends the command line to CP/M-68K. Has the same effect as a carriage return keystroke.
CTRL-P	Echoes all console activity to the printer. You can use CTRL-P after you halt scrolling with CTRL-S. A second CTRL-P ends printer echo. If your system is not connected to a printer, a CTRL-P can cause your system to hang, requiring a cold start, that is a complete resetting of your hardware and software. (Please Section 1.1 for a description of a cold start.)
CTRL-Q	Restarts screen scrolling after a CTRL-S.

Table D-1. (continued)

Keystroke	Action
CTRL-R	Types a # at the current cursor location, moves the cursor to the next line, and retypes any partial command you have typed so far.
CTRL-S	Stops screen scrolling. If a display scrolls by too fast for you to read it, type CTRL-S. CTRL-Q restarts screen scrolling.
CTRL-U	Discards all the characters in the command line that you typed so far, types 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 that you typed so far and moves the cursor back to the beginning of the current line.

End of Appendix D

## Appendix E

### User's Glossary

**ambiguous filename:** Filename that contains either of the CP/M-68K wildcard characters, ? or \*, in the primary filename or the filetype or both. When you replace characters in a filename with these wildcard characters, you create an ambiguous filename and can easily reference more than one CP/M-68K file in a single command line. See Section 2 of this manual.

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

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

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

**attribute:** File characteristic that can be set to on or off.

**back-up:** Copy of a disk or file made for safe keeping, or the creation of the 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 reserved for a specific use.

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

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

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

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

**command:** Elements of a CP/M-68K command line. In general, a CP/M-68K command has three parts: the command keyword, the command tail, and a carriage return.

**command file:** Series of coded machine executable instructions stored on disk as a program file, invoked in CP/M-68K by typing the command keyword next to the system prompt on the console. The CP/M-68K command files generally have a filetype of 68K. Files are either command files or data files. Same as a command program.

**command keyword:** Name that identifies an CP/M-68K command, usually the primary filename of a file of type 68K, or a built in command. The command keyword precedes the command tail and the carriage return in the command line.

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

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

**concatenate:** Term that describes one of PIP's operations that copies 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:** Nonprinting character combination that sends a simple command to CP/M-68K. Some control characters perform line editing functions. To enter a control character, hold down the CONTROL key on your terminal 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:** Nonexecutable collection of similar information that generally requires a command file to manipulate it.

**default:** Currently selected disk drive and user number. Any command that does not specify a disk drive or a user number references the default disk drive and user number. When CP/M-68K is first invoked, the default disk drive is drive A, and the default user number is 0, until changed with the USER command.

**delimiter:** Special characters that separate different items in a command line. For example, in CP/M-68K, a colon separates the drive specification from the filename. A period separates the filename from the filetype. Brackets separate any options from their command or file specification. Commas separate one item in an option list from another. All of the above special characters are delimiters.

**directory:** Portion of a disk that contains entries for each file on the disk. In response to the DIR command, CP/M-68K displays the filenames stored in the directory.

**DIR attribute:** File attribute. A file with the DIR attribute can be displayed by a DIR command.

**disk, diskette:** Magnetic media used to store information. Programs and data are recorded on 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. CP/M-68K assigns a letter to each drive under its control. For example, CP/M-68K may refer to the drives in a four-drive system as A, B, C, and D.

**editor:** Utility program that creates and modifies text files. An editor can be used for creation of documents or creation of code for computer programs. The CP/M-68K editor is invoked by typing the command ED next to the system prompt on the console. See ED in Section 6 of this manual.

**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:** 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, filename, filetype and other information describing a file to be accessed or created on the disk.

**filename:** Name assigned to a file. A filename can include a primary filename of 1-8 characters and a filetype of 0-3 characters. A period separates the primary filename from the filetype.

**file specification:** Unique file identifier. A complete CP/M-68K file specification includes a disk drive specification followed by a colon (:), a primary filename of 1 to 8 characters, a period and a filetype of 0 to 3 characters. For example, b:example.tex is a complete CP/M-68K file specification.

**filetype:** Extension to a filename. A filetype can be from 0 to 3 characters and must be separated from the primary filename by a period. A filetype can tell something about the file. Certain programs require that files to be processed have certain filetypes. See Appendix C.

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

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

**hardware:** Physical components of a computer.

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

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

**input:** Data going into the computer, usually from an operator typing at the terminal or by a program reading from the disk.

**interface:** Object that allows two 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.

**logged in:** Made known to the operating system, in reference to drives. A drive is logged-in when it is selected by the user or an executing process, and remains selected or logged in until you change disks in a floppy disk drive or enter CTRL-C at the command level.

**logical:** Representation of something that may or may 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**, **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. It enables user-written programs to execute easily.

**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 processor sends to the console or disk.

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

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

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

**primary filename:** First 8 characters of a filename. The primary filename is a unique name that helps the user identify the file contents. A primary filename contains 1 to 8 characters and can include any letter or number and some special characters. The primary filename follows the optional drive specification and precedes the optional filetype.

**program:** Series of specially coded instructions that performs specific tasks when executed by a computer.

**prompt:** Any characters displayed on the video screen to help the user decide what the next appropriate action is. A system prompt is a special prompt displayed by the operating system. The system prompt indicates to the user that the operating system is ready to accept input. The CP/M-68K system prompt is an alphabetic character followed by an angle bracket. The alphabetic character indicates the default drive. Some applications programs have their own special system prompts.



**Read-Only:** Attribute that can be assigned to a disk file. Every file and drive is either Read-Only or Read-Write. When a file is set to Read-Only, you can read from that file but not write any changes to it. The default setting is Read-Write. Any error in resetting the disk or changing media automatically sets the drive to Read-Only until the error is corrected.

**Read-Write:** Attribute that can be assigned to a disk file. The Read-Write attribute allows you to read from and write to a specific Read-Write file so marked. See **Read-Only**.

**record:** Collection of data. A file consists of one or more records stored on disk. An CP/M-68K record is 128 bytes long.

**RO:** Abbreviation for Read-Only.

**RW:** Abbreviation for Read-Write.

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

**software:** Specially coded programs that transmit machine readable instructions to the computer, as opposed to hardware, 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, text formatter, or assembler.

**syntax:** Format for entering a given command.

**system attribute:** A file attribute. You can give a file the system attribute by using the SYS option in the STAT command. A file with the SYS attribute is not displayed in response to a DIR command; you must use DIRS to view a directory of files having the system attribute. (See Section 5.)

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

**terminal:** See **console**.

**track:** Concentric rings dividing a disk. There are 77 tracks on a typical eight inch floppy disk.

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

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

**user number:** Number assigned to files in the disk directory so that different users need only deal with their own files and have their own directories even though they are all working from the same disk. In CP/M-68K, files can be divided into 16 user 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 match certain specified items. In CP/M-68K there are two wildcard characters, ? and \*. The ? can be substituted for any single character in a filename, and the \* can be substituted for the primary filename or the filetype or both. By placing wildcard characters in filenames, the user creates an ambiguous filename and can quickly reference one or more files.

End of Appendix E

# Index

- \$\$\$ file, 50
  - 68K, 8
  - :n (Through Line Number) Command, 93
- A**
- A (Append) command, 82
  - abort a copy operation, 54
  - access modes, 66-68
  - active user numbers, 70
  - assign physical device to logical device, 71
  - attributes, 67
  - AUXI:, 17
  - AUXO:, 17
  - AXI:, 54
  - AXO, 54
- B**
- B (Beginning/Bottom) command, 87
  - back-up disks, 3-4
  - basic editing commands, 86
  - booting the system, 1
  - bytes, 66
- C**
- C (Character) command, 87, 93
  - character pointer, 85
  - character transfer, 53
  - cold start, 1
  - combined-command line, 93
  - combining files, 52
  - command
    - description, 28
    - keyword, 2, 28
    - line, 2
    - mode, 91
    - tail, 2
  - CON:, 17, 54
  - concatenation, 53
  - control character, 2
  - copy, 3
  - COPY command, 4, 33-36
  - copy files to and from auxiliary devices, 53
  - CP, 85
  - CTRL key, 2
  - CTRL-E, 95
- CTRL-Z, 53, 54, 55, 91, 95
  - cursor, 2
- D**
- D (Delete) Command, 89
  - data files, 7
  - default
    - drive, 9
    - user number, 52
  - dest-filespec, 31
  - DEV:, 71
  - device names, 54
  - dir, 37
  - DIR (Directory) Command, 37
    - 66-68
  - directory, 7, 12
    - attribute, 66
    - entries, 66
    - space, 12
    - verification, 67
  - DIRS (System Directory) Command, 2, 37
  - disk
    - free space, 65
    - space allocation, 67
  - displaying disk status, 69
  - Dn (delete characters) option, 57
  - drive, 12
    - specifier, 9-10, 25
- E**
- E (echo) option, 51, 58
  - E (Exit) Command, 84
  - ED, 7
  - ED Command, 39
  - ED
    - error symbols, 102
    - messages, 80, 94-96, 99, 101
    - prompt, 80, 91
    - syntax, 79
  - editing, 94
  - editors, 79
  - end-of-file character, 53
  - EOF:, 54
  - ERA command, 44

## F

F (filter form-feeds)  
  option, 58  
F (Find) command, 93  
FCB, 66  
file, 7  
  attributes, 50  
  categories, 8  
  concatenation, 53  
  specification, 25, 67  
filename, 25  
filetype, 8, 25

## G

Gn option, 49, 50, 53, 56, 58

## H

H (Head of File) command,  
  84, 94, 98  
H (hex data transfer)  
  option, 58

## I

I (ignore) option, 58  
I (Insert) command, 90  
input devices, 17  
input-output port, 71  
insert mode, 90  
Istring^Z (Insert String)  
  Command, 91

## J

J (Juxtapose) command, 95

## K

K (Kill) command, 89  
kilobytes, 66

## L

L (Line) command, 87, 93  
L (translate case) option, 58  
line editing controls, 91  
  characters, 17  
line numbers, 82  
loading CP/M-68K, 1  
logical device names, 17, 54, 71  
long form of PIP, 51  
LST:, 17, 54

## M

M (Macro) command, 96  
memory buffer, 80, 82  
multiple command mode, 55  
multiple file copy, 51

## N

N (add line numbers) option, 58  
n (Number) command, 88  
N Command, 94  
n: (Line Number) Command, 92  
NUL:, 54  
numeric argument, 85  
O (object file transfer)  
  option, 53, 58  
O (Original) Command, 98  
on-line  
  disk, 12  
  status, 69  
Options, 28  
output devices, 17

## P

P (Page) Command, 92  
peripheral devices, 17  
physical device  
  drivers, 71  
  names, 70  
PIP, 7, 30  
  command, 49  
  messages, 52  
  options, 50, 56  
Pn (set page length) option, 60  
pound sign argument, 85  
PRN:, 54  
program file, 1, 7

## Q

Q (Quit) command, 99  
Qs (quit copying) option, 60

## R

R (read system files)  
  option, 60  
R (Read) command, 97  
Read-Only  
  attribute, 64, 66  
  files, 50  
Read-Write attribute, 64, 66  
ready status, 12

real file size, 67  
records, 66  
recovering from editing error  
  conditions, 101  
recs, 66  
REL, 8  
REN  
  command, 62  
  messages, 63  
repeated execution of ED  
  commands, 96  
reset, 13  
RETURN key, 2  
RO, 64-68  
RW, 64-68

## S

S (Substitute) command, 92  
set file access modes, 68  
short form of PIP, 50  
single file copy, 49  
SIZE, 67  
source file, 80, 82  
square brackets, 50, 56  
Ss (start copying) option, 59  
STAT command, 64-67  
syntax notation, 28-30  
System (SYS) attribute, 66-68  
system  
  prompt, 2, 13  
  reset, 2

## T

T (Type) Command, 88, 93, 94  
tab characters, 77  
temporary file, 50, 80  
text editor, 7  
Text Transfer Commands, 82  
Tn (expand tabs) option, 60  
Type, 28  
TYPE command, 77

## U

U (translate case) option, 60  
U (upper-case translation)  
  command, 91  
USER command, 52, 78  
user numbers, 70, 78

## V

V (verify) option, 51, 60  
VAL:, 70  
version number, 1  
virtual file size, 67

## W

W (write over) option, 60  
W (write) command, 84  
W option, 50  
wildcard characters, 11, 26, 66

## X

X (Transfer) Command, 96  
X\$\$\$\$\$\$\$.LIB file, 97

## Z

Z (zero the parity bit) option, 60