

In Figure 2-4, you are reopening the file "REPORT.DOC" to edit it. First you enter **ED REPORT.DOC** to invoke ED. Then you must enter ED's **A** command to bring a copy of a specified number of text lines from the disk into the buffer. The disk copy of "REPORT.DOC" remains on the disk.

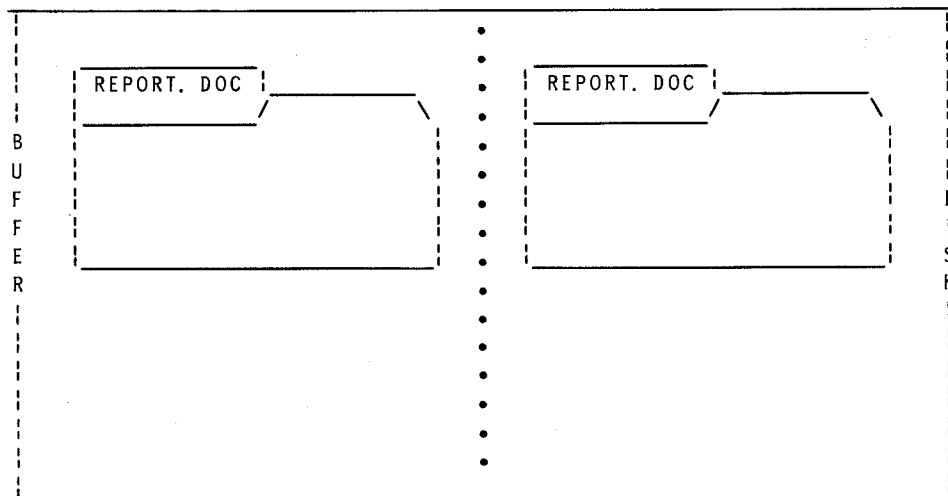


Figure 2-4

You changed and/or added to the text lines that were brought into the buffer. To save these changes and/or additions, you again enter a CTRL-Z and ED's **W** command. Figure 2-5 reflects the state of the files after these activities. The original disk copy of "REPORT.DOC" remains on the disk, a copy of some of the files text remains in the buffer under the name "REPORT.DOC", and some number of edited text lines from the file have been sent to the temporary disk file named "REPORT. \$\$\$".

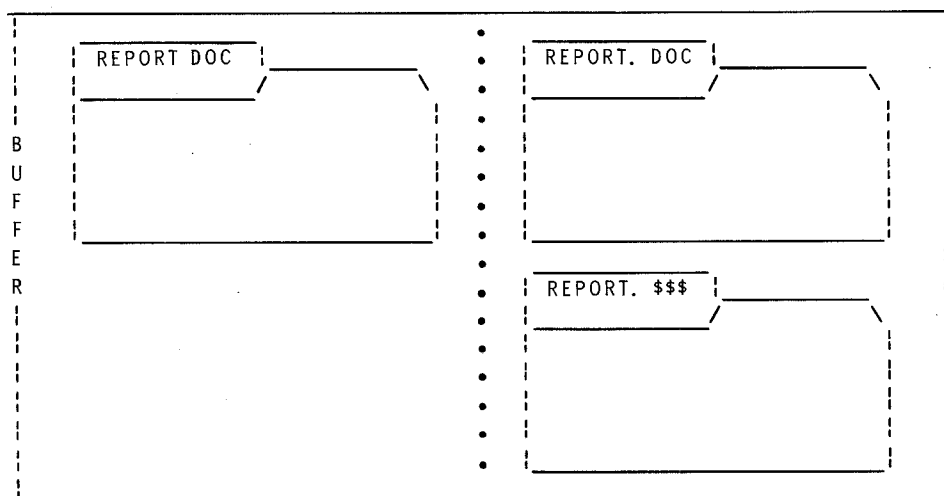


Figure 2-5

In Figure 2-6, you have closed the file editing session and saved the change and/or additions made to the text, by entering ED's E command. ED reacts to this entry by combining all text from buffer file "REPORT.DOC", all text from temporary disk file "REPORT. \$\$\$", and any unchanged text from the original disk file "REPORT.DOC" into a new disk file named "REPORT.DOC". Then ED renames the original "REPORT.DOC" disk file to "REPORT.BAK". The buffer file and the temporary disk file disappear.

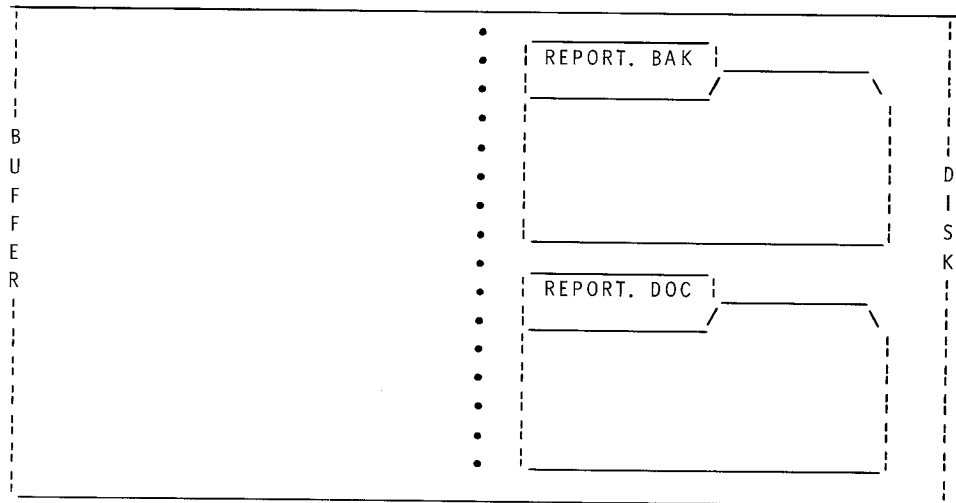


Figure 2-6

Thus a file with a file name extension assigned by you is usually the latest version of the file, whereas a file with the extension "BAK" is usually an old copy of the file.

If you re-edit the disk file "REPORT.DOC" and save the newly-edited version, the current disk file "REPORT.DOC" will become the new disk file "REPORT.BAK", as the memory file "REPORT.DOC" and the disk file "REPORT. \$\$\$" combine to become the new disk file "REPORT.DOC". The old disk file "REPORT.BAK" will be automatically deleted after the third ED session with the file.

NOTE: These diagrams only illustrate a few of the basic options for file manipulation. The text on ED commands explains several other options.

## 5 ED ERROR MESSAGES

```
BREAK"? " AT x  
: *
```

EXPLANATION: (“x” is an invalid character that you entered.) you entered an ED command under inappropriate circumstances, at the wrong kind of prompt, or with improper syntax. Command should be re-entered.

```
DISK OR DIRECTORY FULL
```

EXPLANATION: You either entered **ED** without a file name argument at the CP/M system prompt, or composed more text than the disk could hold. For the former, you should invoke the ED utility by typing a command line with a file name argument. For the latter, you cannot save the overflow of text. When composing future documents, you should use ED’s “W” command or CP/M’s STAT utility more often.

```
** FILE IS READ/ONLY **
```

EXPLANATION: You tried to save newly-edited text to a file that cannot be written to because it has read/only status. You should abandon the text in the buffer with ED’s “Q” exit command, and then use CP/M’s STAT command to assign read/write status to the file before the next edit.

```
"SYSTEM" FILE NOT ACCESSIBLE
```

EXPLANATION: You tried to edit a file that had been given the “system” status by the STAT command. This status hides a file from commands such as ED and DIR. You should assign the “directory” status to the file, using the STAT command, before trying to edit the file again.

```
NO MEMORY
```

EXPLANATION: You have filled memory buffer, and should use the “W” command to send some of the buffer text to the disk.



# ERA

## *The Resident Command for Erasing Files*

The ERA resident command frees space once occupied by disk files, thereby permitting the storage of new files. ERA will erase a single file (1) or a group of files (2). ERA also displays a message when it cannot erase a file (3). Caution should be exercised when using ERA because erased files cannot be recovered under most circumstances.

### 1 ERASING A DISK FILE

To erase a file from a disk, you should enter a command line in the following form:

```
A>ERA {filename.ext} RETURN
```

Where {filename.ext} is the complete name of the file you want to delete.

Files that do not reside on a disk in the default drive can only be erased when their file name is preceded in the command line by the appropriate drive specification.

## 2 ERASING GROUPS OF FILES

A group of files with similar names can be deleted by a single ERA command line when ambiguous file names (names with the "\*" or "?" characters) are used, as in the following example:

```
A>ERA B:PROGRAM?.PRN RETURN
```

which would delete files with names such as PROGRAM1.PRN, PROGRAM2.PRN, PROGRAM7.PRN, and PROGRAM/.PRN from the disk in drive B.

The following command example shows how to delete even more files at once:

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

This entry would erase every file from a disk. Because of the destructive potential of this form of the command, erase will ask you for confirmation with this message:

```
ALL FILES (Y/N)?
```

The command will not be executed until you confirms it by pressing the "Y" key. If the "N" key is pressed, the system prompt will be displayed.

### 3 CONSOLE RESPONSE TO ERA COMMANDS

When ERA finds the specified file and erases it, the system prompt returns.

If the specified file does not reside on a disk in the specified drive, then the console will display the following message:

```
NO FILE
```

If the file you desire to erase is write protected by the "R/O" (read only) status, or if you switched disks between drives without performing a warm boot, a message like the following will be displayed:

```
Bdos Err On x: File R/O
```

Where "x" is the letter of the drive containing the write-protected file.

If the disk containing the file to be erased is mechanically write protected (with adhesive tabs for 5.25-inch, without adhesive tabs for 8-inch, or with disk drive cabinet switches for 8-inch and Winchester Disks), then a message in the following form will be displayed:

```
Bdos Err On x: Bad Sector
```

Where "x" is the drive from which you tried to erase a file.

If you specified a drive in an ERA command, and that drive contains no disk, then a message in the following form will be displayed:

```
Bdos Err On x: Select
```

Where "x" is the drive from which you tried to erase a file.





# FORMAT

## *The Utility that Prepares the Disk Surface*

FORMAT prepares a floppy disk or a Winchester Disk partition for the storage of data by establishing storage areas on the disk surface. At the same time, FORMAT erases any data that remains on the disk from prior use, and sometimes inspects the recording surface for imperfections that could impair data storage or transmission. FORMAT also enables you to determine how much data you will be able to store on the disk.

**CAUTION:** Because FORMAT erases all existing data on a disk, make certain that you only format blank disks or disks containing expendable data. You can use the DIR (refer to the text titled “DIR”) or STAT (refer to the text titled “STAT”) commands to check a disk for valuable data files before formatting it. Notice, however, that the DIR and STAT commands cannot always find all of the data on a disk.

You can use the FORMAT utility through either of two methods: the FORMAT Prompt Method or the System Prompt Method.

## **1 FORMAT PROMPT METHOD**

With this FORMAT method, you load the FORMAT utility into memory by entering a command at the system prompt. Then you answer a series of FORMAT prompts to define the formatting operation.

## 1.1 FORMAT Invocation

Answer the system prompt with a command in the following form:

A>**FORMAT RETURN**

When invoked through the FORMAT prompt method, FORMAT first identifies itself with name, version number, and a caution about its capabilities. Then it asks you if you wish to continue the operation, as shown:

```
Format Version 2.04
This program is used to initialize a disk.
All information currently on the disk will be destroyed.
Is that what you want? (y/n):
```

If you wish to prepare a disk, enter a Y at this prompt and continue reading.

If you do **not** wish to prepare a disk, enter any character other than a Y. Control will return to the operating system, which will display a system prompt.

## 1.2 Specifying The Disk to be Formatted

After you have confirmed your intention to format a disk, FORMAT asks:

```
Which drive do you wish to use for this operation?:
```

Answer this prompt by entering the letter of the drive containing the disk you wish to prepare.

## 1.3 Disk Format Specification

Heath/Zenith disk drive models accommodate four different kinds of disk:

- 5.25-inch hard-sectored floppy disks;
- 5.25-inch soft-sectored floppy disks;
- 8-inch floppy disks; and
- Winchester Disk

Each of these disks types are formatted differently. Once you have specified your drive, the messages and prompts that FORMAT displays depend on the type of the disk you are formatting.

### FORMATTING A 5.25-INCH HARD-SECTORED FLOPPY DISK

After you have selected a drive for this FORMAT operation, FORMAT prompts you as follows:

```
Put the disk you wish to be formatted in drive x.  
Press RETURN to begin, anything else to abort.
```

Entering a carriage return at this prompt will begin the actual formatting operation, while entering any other keyboard character will cause all FORMAT activities to end as CP/M displays the system prompt.

When the operation is finished, FORMAT will display:

```
Do you have any other disks to be formatted? (y/n):
```

After formatting, this disk has a file capacity of 90 kilobytes.

### FORMATTING A 5.25-INCH SOFT-SECTORED FLOPPY DISK

After you select a drive for this FORMAT operation, FORMAT displays the prompt:

```
Which density? (S=single, D=double):
```

The “density” of a disk refers to the concentration of data on its surface. You can decide whether data will be stored on the disk at “single” density or “double” density by entering “S” or “D” at this prompt.

If you specify "D" for "double" density, data will be recorded on the disk at approximately twice the concentration as it would at "single" density. Higher density levels might also decrease data access reliability.

After the density is selected, FORMAT will display the prompt:

```
Number of Sides? (1=single, 2=double):
```

Since 5.25-inch soft-sectored disks have two usable sides, you have the option of selecting how many of these sides will be prepared to store data.

After you enter a valid response to each of these prompts, FORMAT will display either of the following two messages:

```
48 TPI drive -- 40 tracks will be formatted
```

```
Put the disk you wish to be formatted in drive x.  
Press RETURN to begin, anything else to abort.
```

OR

```
96 TPI drive -- 80 tracks will be formatted
```

```
Put the disk you wish to be formatted in drive x.  
Press RETURN to begin, anything else to abort.
```

Entering a carriage return at this prompt will begin the actual formatting operation, while entering any other keyboard character will cause all FORMAT activities to end as CP/M displays the system prompt.

When the operation is finished, FORMAT will display the message:

```
Do you have any more disks to format? (y/n):
```

NOTE: Soft-sectored 5.25-inch disks supported by the Heath/Zenith CP/M are available in two varieties: those with 48 tracks per inch (48 tpi) and those with 96 tracks per inch (96 tpi). Each variety has different storage capacities, depending on the density and number of sides you select when formatting.

The following table shows the file capacities of 5.25-inch, soft-sectored, 48 tpi disks:

	Single-sided	Double-sided
Single density	90 kilobytes	190 kilobytes
Double density	148 kilobytes	308 kilobytes

The following table shows the file capacities of 5.25-inch, soft-sectored, 96 tpi disks:

	Single-sided	Double-sided
Single density	190 kilobytes	388 kilobytes
Double density	308 kilobytes	624 kilobytes

NOTE: The CP/M Operating System is set to accommodate either 48 tpi disks or 96 tpi disks in a Z-37 drive model. This setting can be adjusted through the CONFIGUR utility. (See "Submenu B: Set Disk Parameters" in the text on CONFIGUR.)

### FORMATTING AN 8-INCH DISK

After you select the drive to be used to FORMAT this disk, the FORMAT utility will display one of the following prompts:

Which density? (S=single, D=double, E=extended double):

OR

Which density? (S=single, D=double):

The "density" of a disk refers to the concentration of data on its surface. You can decide whether data will be stored on the disk at "single" density or "double" density or "extended double" density by entering "S" or "D" or "E" at this prompt. If you specify "D" for "double" density, data will be recorded on the disk at about twice the concentration as it would at "single" density. Extended double density is the highest level of data concentration possible on that disk. Higher density levels might also decrease data access reliability.

After you answer the density prompt, FORMAT displays:

```
Put the disk you wish to be formatted in drive x.
Press RETURN to begin, anything else to abort.
```

Entering a carriage return here will begin the formatting operation, while entering any other keyboard character will cause all FORMAT activities to end as CP/M displays the system prompt.

When the operation is finished, FORMAT will display the message:

```
Do you have any more disks to format? (y/n):
```

Heath/Zenith CP/M supports two varieties of 8-inch disk: single-sided and double-sided. Each variety has different storage capacities, depending on the density and number of sides you select during formatting.

The following table shows the file capacities of single-sided 8-inch disks:

Single density	Double density	Extended double density
241 kilobytes	482 kilobytes	596 kilobytes

The following table shows the file capacities of double-sided 8-inch disks:

Single density	Double density	Extended double density
490 kilobytes	980 kilobytes	1208 kilobytes

NOTE: The 8-inch floppy disk drive slot of the H/Z-67 drive model cannot be used to FORMAT a disk to extended double density. Furthermore, an 8-inch disk that is formatted to extended double density cannot be used in the H/Z-67 model.

## FORMATTING A WINCHESTER DISK PARTITION

After you specify the drive name assigned to the partition, FORMAT displays the prompt:

```
Press RETURN to begin, anything else to abort.
```

Entering a carriage return here will begin the formatting operation, while entering any other keyboard character will cause all FORMAT activities to end as CP/M displays the system prompt.

When the formatting operation begins, the following message is displayed:

```
Formatting partition
```

When the operation is finished, FORMAT will display the message:

```
Do you have any more disks to format? (y/n):
```

### 1.4 Ending a FORMAT Operation

When FORMAT finishes preparing the surface of a disk or partition, it will display the following prompt:

```
Do you have more disks to format? (y/n):
```

The specified disk or partition is now ready for data storage.

If you wish to FORMAT another disk or partition immediately, press **Y** at this prompt.

If you have no other disks to FORMAT, press **N** at this prompt and FORMAT will display the following message:

```
Place a bootable disk in drive x and press any character:
```

Where “x” is the letter for the default drive.

If the disk that was used to perform bootstrap has been removed from drive A since bootstrap, then it should be inserted back into drive A now. The entry of any keyboard character will transfer control back to the operating system, which will display a system prompt.

## 2 SYSTEM PROMPT METHOD

The System Prompt Method enables you to include all of the specifications necessary for the FORMAT operation in a single command line entered at the CP/M system prompt.

### 2.1 Command Line Entry

FORMAT commands that are entered by the System Prompt Method should be entered in the following form:

```
A>FORMAT {drive}:{[option,option]} RETURN
```

Where **FORMAT** is the command line function

where **{drive}** is the letter of the drive that contains the disk you wish to format (this letter must represent a valid drive in your hardware environment, such as **A, B, C, D, E, or F**); and

where **{[option,option]}** represents letters and/or numbers enclosed in square brackets [] and separated by commas , to specify how the formatting operation should be conducted.

### 2.2 FORMAT Options

FORMAT command lines entered by the System Prompt Method can include the following options:

- SD** Disk formatted to Single Density.
- DD** Disk formatted to Double Density.
- 1S** One Side of disk formatted.
- 2S** Two Sides of disk formatted.
- F** Fast formatting, because the routine test of disk surface media is not performed.



- N** No prompt displayed between **FORMAT** command entry and **FORMAT** execution.
- T1** Type of disk to be formatted must be a 5.25-inch hard-sectored disk.
- T3** Type of disk to be formatted must be a 5.25-inch soft-sectored disk.
- T4** Type of disk to be formatted must be an 8-inch disk, within a Z47 or H47 drive model.
- TH6** Type of "disk" to be formatted must be a partition on the Winchester Disk within a Z67 drive model. This partition must have been assigned a drive name by the **ASSIGN** utility.
- TF6** Type of disk to be formatted must be an 8-inch disk, within the floppy disk slot of a Z67 drive model.

**NOTE:** It is not necessary to enter an option for disk type (**T1**, **T3**, **T4**, **TH6**, **TF6**) in a System Prompt Method **FORMAT** command, because **FORMAT** automatically checks the type of the disk within the specified drive.

However, disk type should be specified when you desire to format a disk **only** if it is of a particular type. If a disk type is specified as an option, and the drive specified for this operation contains a disk of a different type, then the following error message is displayed:

DISK IS NOT OF TYPE SPECIFIED

## 2.3 FORMAT Defaults

When you enter a FORMAT command line with a drive specification, and neglect to specify some or all of the possible options, FORMAT will prepare the disk according to the following default criteria:

- Disk formatted to Single Density (as if you specified the SD option);
- One side of disk formatted (as if you specified the 1S option);
- Disk surface will be tested for data retention (as if you did **not** specify the F option); and
- Prompts will be displayed between FORMAT command entry and FORMAT execution (as if you did **not** specify the N option)

Unless the N option is entered in the command line, the following prompt will be displayed after you enter the FORMAT command:

```
Format Version 2.04
This program is used to initialize a disk.
All information currently on the disk will be destroyed.
Is that what you want? (y/n):
```

To confirm your intention to run a FORMAT operation, enter a **Y** at this prompt. FORMAT will display a prompt in the form of the example below.

To abort the FORMAT utility at this point, enter any other keyboard character. The operating system will take control and display a system prompt.

```
Put the disk you wish to be formatted in drive x.
Press RETURN to begin, anything else to abort.
```

To begin execution of the FORMAT operation, enter a carriage return at this prompt.

To abort the FORMAT utility, enter any other keyboard character. The operating system will take control and display a system prompt.

## 2.4 System Prompt Method Examples

### A>FORMAT B: RETURN

FORMAT will prepare the surface of the disk in drive B. By default, this disk will be formatted to single density on only one side. Also by default, FORMAT will display prompts before formatting, and test the disk surface while formatting.

### A>FORMAT B:[DD,2S,T4] RETURN

FORMAT will prepare the surface of the disk in drive B to double density and on both sides, as specified by options. FORMAT will display prompts before formatting and test the disk surface while formatting, by default. However, if drive B is not a Z47 or H47 drive containing an 8-inch disk, an error message will be displayed and no formatting operation will occur.

### A>FORMAT B:[SD,1S,DD,2S] RETURN

If your command line contains contradictory options, FORMAT will acknowledge the last one. Hence, in this case, FORMAT will prepare the surface of the disk in drive B to double density, as specified by the last density option (DD), and on both sides, as specified by the last side quantity option (2S). FORMAT will display prompts before formatting and test disk surface while formatting, by default.

### A>C:format B:[Dd,f,N] RETURN

The FORMAT utility, in this case, is stored on the disk in non-default drive C. It will prepare the surface of the disk in drive B to double density, as specified by the "Dd" option. But since no option is specified for the number of sides, only one side will be formatted (by default). The "f" option specifies that this formatting operation will be performed without a disk media test. The "N" option specifies that FORMAT will not prompt you to confirm your intentions before the formatting operation begins.

NOTE: FORMAT command lines can be edited (by pressing the **DELETE** key) or erased entirely (by holding down the **CTRL** key and pressing the **X** key).

### 3 FORMAT ERROR MESSAGES

Drive not available in current configuration.

**EXPLANATION:** If you entered a drive name that does not exist in the hardware environment, then a different drive name should be entered. If you entered a drive name for which your copy of the CP/M Operating System has not yet been customized, then MAKEBIOS or CONFIGUR must be run before FORMAT will accept the drive name entry. If you are using a Z67 drive model and entered the drive name for a partition that has not yet been established by the ASSIGN utility, then ASSIGN must be run before FORMAT.

Unable to format this disk. It is write protected.  
Do you have any more disks to format? (y/n):

**EXPLANATION:** If the disk is a 5.25-inch disk you should remove the adhesive tab from the notch on the disk cover. If the disk is an 8-inch disk the you should cover the notch on the disk cover using the tabs provided, and push down the write protect switch on the front of the H/Z47 or H/Z67 cabinet.

Unable to format this disk. Place a different disk in the drive and press any key to begin...

**EXPLANATION:** The disk to be formatted is damaged or improperly inserted in the drive. You should try the operation again and replace the disk if the message appears again.

Media error

**EXPLANATION:** The disk to be formatted is damaged or improperly inserted in the drive. You should try the operation again and replace the disk if the message appears again.

Wrong type of media, or media inserted improperly,  
or media damaged.

**EXPLANATION:** You may have tried to FORMAT a hard-sectored disk in a soft-sectored drive, or vice versa. You should check that the proper type of disk is being used. If the proper disk type is being used, then the disk is probably damaged and a different disk should be used for the operation.

ILLEGAL FORMAT OPTION

**EXPLANATION:** System Prompt Method command line was entered with undefined characters in place of options.

## ILLEGAL COMMAND SYNTAX

EXPLANATION: System Prompt Method command line was entered with undefined characters in place of options.

## DISK IS NOT OF TYPE SPECIFIED

EXPLANATION: A System Prompt Method command line specified a drive that contained a disk which did not match the specified disk type.

## OPTION NOT AVAILABLE

EXPLANATION: A System Prompt Method command line included option characters which were not possible under the circumstances. Re-enter command with pertinent options.

Disk is not partitioned. The complete disk surface will be formatted.  
Is that what you want? (y/n):

EXPLANATION: The Winchester Disk has not yet been partitioned. Therefore, any FORMAT operation will erase and prepare the surface of the entire Winchester Disk. To continue FORMAT operation, press Y. If you wish to partition the Winchester Disk before formatting, run the PART utility as instructed in the PART manual.

## PARTITION IS SMALLER THAN MINIMUM ALLOWABLE SIZE

EXPLANATION: Winchester Disk partitions must contain a minimum number of sectors before they can be prepared by FORMAT. Either ASSIGN a partition containing this minimum quantity of sectors, or use the PART utility to repartition the Winchester Disk with few enough partitions so that at least one is of minimum size. Refer to the Z67 manual and the PART manual to determine the minimum allowable size for a partition.

## PARTITION IS LARGER THAN CP/M MAXIMUM SIZE -- ONLY 8 MEGABYTES USABLE

EXPLANATION: Winchester Disk partitions should contain no more than eight megabytes of storage capacity if used with the CP/M Operating System. If a partition contains more, it cannot be formatted. Either ASSIGN a partition of suitable size for formatting purposes, or repartition the Winchester Disk with the PART utility and form a partition within the eight megabyte limit. Then format the properly sized partition.



# LIST

## *The Utility that Prints File Contents on Paper*

The LIST utility enables you to obtain paper copies of files by entering a command (1) for one or more files (2) to be printed. Special printout characteristics can be set when you enters the command with LIST parameters (3). You can stop a LIST printout in progress (4). LIST is used to print out only certain types of files (5).

## **1 METHODS OF ENTERING LIST COMMANDS**

Two different methods can be used to enter LIST commands: the LIST Prompt Entry Method and the System Prompt Entry Method.

## 1.1 LIST Prompt Entry Method

With the LIST Prompt Entry Method, you enter LIST in response to the system prompt. This entry is sufficient to invoke LIST, which displays its own prompt—the asterisk (\*). You can now enter the argument portion of the command line in response to the “\*” prompt supplied by LIST. LIST prompt entries are made in the following form:

**A>LIST RETURN**

**\*{argument} RETURN**

Where {argument} is the name of the file(s) to be listed.

After the LIST operation is finished, LIST again displays the “\*” prompt. You can enter another argument or return to the operating system by entering a carriage return.

## 1.2 System Prompt Entry Method

To invoke a similar operation without splitting up the command line, you must include the argument in the response to the system prompt, as in the following example:

**A>LIST {argument} RETURN**

Where {argument} is the name of the file(s) to be listed.

After LIST finishes the latter printing operation, it automatically returns you to the operating system, and the system prompt is displayed.



## EXAMPLE OF LIST USAGE

The following command, entered using the LIST Prompt Method:

```
A>LIST RETURN
```

```
*REPORT.DOC RETURN
```

and this command, entered using the System Prompt Method:

```
A>LIST REPORT.DOC RETURN
```

will produce the same results. Both commands will produce a paper copy of the file named "REPORT.DOC".

## 2 PRINTING CONTENTS OF MORE THAN ONE FILE

Any number of files can be specified in a single LIST command. To initiate a listing of several files, the names of the files are entered in the argument separated by single spaces.

If the files to be listed reside on different disks, their names should be preceded by a drive specification (drive letter and colon).

Both of the following examples demonstrate how to LIST the contents of the specified files:

```
A>LIST RETURN
```

```
*PRINTOUT.DOC B:PROGRAM.PRN B:REPORT.DOC C:SYSTEMX.PRN RETURN
```

or

```
A>LIST PRINTOUT.DOC B:PROGRAM.PRN B:REPORT.DOC C:SYSTEMX.PRN RETURN
```

### 3 LIST PARAMETERS

You can specify parameters in a LIST argument to alter the characteristics of a standard printout. These parameters allow you to select printout characteristics such as the date, the number of copies desired, the width of tabs, etc. If no parameters are specified, LIST will print a document with default value characteristics.

The parameters used to specify printout characteristics are as follows:

PARAMETER NAME	KEYBOARD ENTRY	DESCRIPTION OF PRINTOUT CHARACTERISTIC	DEFAULT VALUE
Date	<b>[D xxx...x]</b>	First 10 characters of specified date printed on upper left corner of each document page.	none
Heading	<b>[H xxx...x]</b>	First 60 characters of specified heading printed on top line of each document page.	file name
No Heading	<b>[N]</b>	No heading or date printed on any document page.	file name
Lines per Page	<b>[L nn]</b>	Each document page has the specified number of lines. Specification of zero/page lines causes continuous printing.	60 lines per page
Tab Stop Width	<b>[T nn]</b>	Each tab stop within text is expanded or contracted to specified number of spaces.	8 spaces

PARAMETER NAME	KEYBOARD ENTRY	DESCRIPTION OF PRINTOUT CHARACTERISTIC	DEFAULT VALUE
Page Number	[P nn]	Page numbering sequence begins with specified number on first file page.	page 1
Upper Case	[U]	All letters in document printed in upper case.	upper and lower case
Copies Desired	[C nn]	Specified number of copies are printed.	1 copy
Erase	[E]	File is erased from disk after LIST operation is completed.	file retained on disk

### USE OF LIST COMMAND LINE PARAMETERS

Parameters are entered, enclosed in square brackets, after the last file name in the LIST command. If more than one parameter is entered, each should be enclosed in a set of brackets.

A LIST command entered with parameters using the LIST Prompt Method might appear as follows:

```
A>LIST RETURN
*PRINTOUT.DOC PROGRAM.PRN [H Today's Work] [D 31-Feb-81] [P 09] RETURN
```

This command will cause the contents of the files PRINTOUT.DOC and PROGRAM.PRN to be printed, with the following heading across the top of the first page of each file:

Today's Work

31-Feb-81 Page 9

The parameters will take effect on all of the files specified in that command line.

When the command is entered using the LIST Prompt Entry Method, any parameters entered (except for the starting Page Number parameter and Erase parameter) will remain in effect with any files specified at successive asterisk (\*) prompts, until new values are entered for the parameters, or until control is returned to the operating system.

When a LIST command line is entered using the System Prompt Method, letters in the heading, date, and page number line might be automatically translated into upper case. The following System Prompt command demonstrates this character translation:

**A>LIST PRINTOUT.DOC PROGRAM.PRN [H Today's Work] [D 31-Feb-81] [P 09] RETURN**

This command will cause the contents of the files PRINTOUT.DOC and PROGRAM.PRN to be printed with the following heading across the top of the first page of each file:

TODAY'S WORK

31-FEB-81 PAGE 9

## **4 ABORTING A LIST OPERATION**

After a LIST command has been entered, the printout can be aborted by pressing any keyboard character. When you abort a command that was entered by the LIST Prompt Method, the next printout might also be automatically aborted soon after it begins. Therefore, it is advisable to exit from, and reinvoke, the LIST utility after aborting a printout.

## 5 FILES THAT SHOULD BE LISTED

Only files containing ASCII characters should be listed. ASCII character files include files composed using a text editor (such as ED), or file with the "PRN" extension created by an assembler (such as ASM).

Files such as those bearing the extension "COM" or "HEX" will produce meaningless printouts if LISTed because they are not composed of ASCII characters.

Files composed in a word processor will LIST, but they might possess features (such as bold face characters or page breaks) that LIST will not print in the same way as the word processor's printout command.

## 6 LIST ERROR MESSAGES

File not found

**EXPLANATION:** A file specified in the LIST command argument does not exist on the disk in the logged drive. If the file does exist on a disk, that disk should be inserted in a drive and the appropriate drive should be specified before the file name in the argument.

Syntax error in command line

**EXPLANATION:** The LIST command was improperly entered. Often occurs if a space is not entered between file names, or when an invalid parameter is specified.



# LOAD

## *The Utility that Loads a "HEX" File for Execution*

The LOAD utility puts an assembled hexadecimal file in the Transient Program Area (1), and translates the file into a "COM" file, which is executable under the CP/M operating system (2). Only certain types of files can be loaded (3).

### 1 LOAD INVOCATION

To LOAD a hexadecimal file into the Transient Program Area (TPA), you should respond to the system prompt with an entry in the form:

```
A>LOAD {hex file} RETURN
```

Where {**hex file**} is the primary file name of an Intel hexadecimal file. The file extension is omitted from this entry because LOAD always assumed the extension to be "HEX".

This command translates the hex file into machine code that can be executed under CP/M, gives it a "COM" extension, and writes it back to the disk.

## 2 EXECUTING A LOADED PROGRAM

You can start execution of the program by entering the primary name of the file (and a carriage return) in response to the system prompt. Thus the file is executed as if it were a regular CP/M utility.

It is only necessary to LOAD a hex file once. The "COM" version of the file will be stored on the disk from which the "HEX" version came. Thereafter, CP/M treats it like another "COM" file.

The operation can take place on a non-default drive if the file name is prefixed by a drive name. Thus the entry of the following command:

**A>LOAD B:BETA RETURN**

brings the LOAD program into the TPA from the default disk and then operates on the file "BETA.HEX", which resides in drive B. The file named "BETA.COM" is written to the disk in drive B. (The file "BETA.HEX" will also remain on the disk.) Now you can execute BETA.COM by responding to the system prompt as shown:

**A>BETA RETURN**

Thus, you can "invent" new CP/M commands by using LOAD to translate a "HEX" file into a CP/M-executable "COM" file.

## 3 LOAD REQUIREMENTS FOR HEX FILES

To be loaded, a file must contain valid Intel hexadecimal format records. The ASM utility can be used to produce such a file from a file with the "ASM" extension. The hex file must begin at address 100H, which is the memory location at which the Transient Program Area (TPA) begins. In addition, the addresses in the hex records must be in ascending order. Gaps in memory regions are filled with zeroes by the LOAD command, as the hex records are read. Thus, LOAD must be used only for creating CP/M command files, which operate in the TPA. Program files that occupy memory locations other than the beginning of the TPA (address 100H) can be loaded using the DDT utility. After a program is loaded, LOAD displays a message in the following form:

```
FIRST ADDRESS nnnn
LAST ADDRESS nnnn
BYTES READ nnnn
RECORDS WRITTEN nn
```



## 4 LOAD ERROR MESSAGES

INVALID HEX DIGIT

EXPLANATION: The hexadecimal file you tried to LOAD contains upper bits set in ASCII words.

ERROR ADDRESS nnnn

EXPLANATION: An error occurred at address "nnnn".

CHECK SUM ERROR

EXPLANATION: The hexadecimal file did not produce the correct check sum during LOAD execution.

ERROR: CANNOT OPEN SOURCE, LOAD ADDRESS nnnn

EXPLANATION: The hexadecimal source file specified in the LOAD command line is not present on the specified disk.

NO MORE DIRECTORY SPACE

EXPLANATION: The directory on the disk is full.

CANNOT CLOSE FILES

EXPLANATION: The loaded file is not present when LOAD tries to close the "COM" file.

INVERTED LOAD ADDRESS, LOAD ADDRESS nnnn

EXPLANATION: The loaded hexadecimal file did not start at beginning of the TPA (0100H) as it should have.



# MAKEBIOS

## *The Utility that Customizes the Operating System for Disk Drives*

The MAKEBIOS utility helps you to create the part of CP/M known as the Basic Input/Output System (BIOS). The kind of BIOS that MAKEBIOS creates is selected by you to accommodate any combination of disk drives offered by Heath/Zenith. You can implement this utility by running MAKEBIOS (1) and selecting a drive combination (2). MAKEBIOS execution is automatic (3).

NOTE: The BIOS is a component of the CP/M Operating System that makes the system work on Heath/Zenith hardware. The file "BIOS.SYS" (which is included with your CP/M software) contains the BIOS.

## 1 RUNNING MAKEBIOS

After making a backup copy of the CP/M Distribution Disk, you can run MAKEBIOS by entering a command line in the form:

**A>SUBMIT MAKEBIOS {destination drive};{file name} {source drive} RETURN**

Where **{destination drive}** is the name of the drive/partition device to which you wish to send the customized BIOS;

where **{file name}** is the optional name of the customized BIOS. If no name is entered, the file containing the tailored BIOS will be called "BIOS.SYS". Any file on the destination disk previously named "BIOS.SYS" will be overwritten.;

where **{source drive}** is the optional name of the drive/partition upon which the following files reside: MAKEBIOS.COM, MAKEBIOS.SUB, PREL.COM, BIOS.ASM; and

where the following files must reside on the default drive: ASM.COM and SUBMIT.COM.

An example of such a command line would be as follows:

**A>SUBMIT B:MAKEBIOS C: B: RETURN**

NOTE: Neither the default disk nor the disk containing the MAKEBIOS files can be write protected during a MAKEBIOS run.

## 2 SELECTING THE APPROPRIATE DRIVE COMBINATION

The entry of the preceding example command line will first produce the display of a menu listing assortments of Heath/Zenith drive controllers:

```
A>MAKEBIOS A:1 C:
```

```
BIOS SELECTION MENU
```

```
A -- H17 ONLY  
B -- H37 ONLY  
C -- H47 ONLY  
D -- H67 ONLY  
E -- H17 AND H37  
F -- H17 AND H47  
G -- H17 AND H67  
H -- H37 AND H47  
I -- H37 AND H67  
J -- H47 AND H67
```

```
ENTER SELECTION:
```

You should enter the selection letter corresponding to the disk drive combination being used.

**NOTE:** The above selections refer to disk drive controllers. Most disk drive models bear the same number as that of the controller that controls them, except that the "H-17" controller can control the H-77 drive unit, the H/Z-87 drive unit, or the H/Z-17 drive within the H/Z-89 micro-computer model.

### 3 MAKEBIOS EXECUTION

From this point on, MAKEBIOS execution is automatic. No further user participation is required.

After you enter a letter at the "ENTER SELECTION:" prompt, a series of command lines and execution messages will appear on the screen. The following example shows the form of this display. (Numeric values will vary.)

```
A>ASM BIOS. AAZ
CP/M ASSEMBLER - VER 2.0
1537
02DH USE FACTOR
END OF ASSEMBLY
```

```
A>REN A: BIOS. HX0 = BIOS. HEX
A>MAKEBIOS A: 2
```

```
A>ASM BIOS. AAZ
CP/M ASSEMBLER - VER 2.0
1637
02DH USE FACTOR
END OF ASSEMBLY
```

```
A>REN A: BIOS. HX1 = BIOS. HEX
A>PREL A: BIOS C: BIOS
```

```
A>REN C: BIOS. SYS = BIOS. HX1
A>MAKEBIOS C: 3
```

```
MAKEBIOS FUNCTION COMPLETE
```

When the display is complete, the destination drive will contain a BIOS suitable for your disk drive combination. This BIOS will bear the file name "BIOS.SYS", or whatever name you might have specified.

This new file will now help control input/output operations for your hardware. It may also be transferred to other disks or partitions, using the PIP utility with the "[R]" parameter.

## 4 MAKEBIOS FUNCTION

The CP/M Operating System requires a Basic Input/Output System (BIOS) to control input/output operations for any peripheral devices that are connected to the Heath/Zenith computer. Among these devices may be the disk drives, a keyboard, a video monitor and a printer.

Your CP/M Distribution Disk (or disk set) comes with a BIOS that will adequately control one of your disk drive types. However, Heath/Zenith offers four different kinds of disk drive, and each hardware environment can include any combination of one or two different drive types. Because a BIOS capable of controlling all combinations would take up too much memory space, the MAKEBIOS files are provided to enable to create a BIOS that will control your own disk drive combination.

Therefore, users with two different kinds of disk drive should run MAKEBIOS.

## 5 A PERSPECTIVE ON MAKEBIOS INVOCATION

You do not invoke the MAKEBIOS.COM utility directly. You enter a SUBMIT command, which triggers the automatic entry of several command lines. The command lines that invoke MAKEBIOS.COM and other CP/M commands are stored in the file named "MAKEBIOS.SUB".

The utilities and resident commands specified within the MAKEBIOS.SUB file automatically modify, assemble, and relocate the source file BIOS.ASM. After these activities are performed, the assembled BIOS file is written to the specified file name (or default name BIOS.SYS) on the specified drive.

## **6 FURTHER BIOS MODIFICATIONS**

You can perform further modifications to the BIOS before running MAKEBIOS by carefully altering the source file BIOS.ASM, which is included with your CP/M distribution media.

However, BIOS.ASM should not be altered until after the the distribution disk has been copied, and the MAKEBIOS.SUB file has been SUBMITTED.

As noted within the BIOS.ASM program, no alterations should be made to the first five lines of the program under any circumstances. The first five lines of the program affect subsequent locations in the program, and user alterations to them could easily render the entire program useless.

## **7 MAKEBIOS ERROR MESSAGE**

ERROR IN EXECUTION OF MAKEBIOS

EXPLANATION: Check the parameters in the SUBMIT MAKEBIOS command line, check the arrangement of the necessary disk files, and run MAKEBIOS again.



## **MOVCPM<sub>xx</sub> (MOVCPM17, MOVCPM37, MOVCPM47, and MOVCPM67)**

*The Utilities that Customize a CP/M Operating System  
Kernel to Fit Memory Size and Disk Type*

After selecting the proper MOVCPM utility to use (1), you can implement it to adjust the system kernel so that it has the proper memory limit and will work on disks of a particular type (3). This utility should be followed immediately with another utility or command (5).

## 1 SELECTING THE PROPER MOVCPM UTILITY

MOVCPM17 is run when you wish to move a system kernel to a 5.25-inch, hard-sectored disk.

MOVCPM37 is run when you wish to move a system kernel to a 5.25-inch, soft-sectored disk.

MOVCPM47 is run when you wish to move a system kernel to an 8-inch disk that will be used to perform bootstrap in an H/Z47 drive model.

MOVCPM67 is run when the you wish to move a system kernel to a Winchester Disk partition, or to an 8-inch disk that will be used to perform bootstrap in an H/Z67 drive model.

In this text section, MOVCPMxx will be used to indicate any of these four utilities designed for use with CP/M Version 2.2.04., where "xx" is the model number of the drive unit receiving the system kernel.

## 2 FUNCTION OF MOVCPMxx

MOVCPMxx loads the kernel of a CP/M Operating System (the part exclusive of the file BIOS.SYS) into a special location in computer memory. At this location, it adjusts the system kernel to either a specified memory size or the total available memory size of the computer. MOVCPMxx must also observe and measure the BIOS.SYS file that will eventually be used with the system kernel, in order to allow sufficient space for the BIOS.SYS file. In addition, MOVCPMxx must make adjustments for the type of disk to which the system kernel will eventually be transferred. MOVCPMxx, however, will rely upon the SYSGEN utility to copy the system kernel that MOVCPMxx loaded into memory.

### 3 MOVCPM<sub>xx</sub> COMMAND LINE ENTRY

The MOVCPM<sub>xx</sub> command line is entered in the following form, with one mandatory specification, two optional specifications and a space separating each, as shown:

**MOVCPM<sub>xx</sub> nn z RETURN**

Where **xx** is the number that identifies a particular MOVCPM utility being used. It also stands for the type of disk drive that will be the destination of the system kernel (17, 37, 47, or 67). One of these numbers is a mandatory component of any MOVCPM command;

where the **nn** variable represents the memory size the transferred system kernel will occupy, in multiples of 1024 bytes (kilobytes). This is an optional value. If the "\*" character or no value is entered, the system kernel will be set to occupy the entire memory capacity of the computer being used, by default. (H/Z89 computers have RAM capacities of 32K, 48K, or 64K.) This value can be less than or equal to the actual memory capacity of the computer. It should not be greater than the memory capacity of the computer.

where the **z** variable represents the location of the BIOS that is to be matched up with the system kernel being moved. You should enter the name of the drive that contains the appropriate BIOS. If the you has renamed the BIOS to be used (by using the MAKEBIOS or REN commands), then this variable must include the name of the drive containing this BIOS and the name of the file containing the BIOS. This variable is optional. If omitted, the MOVCPM utility will assume that the created system kernel will be matched with a copy of the BIOS that was loaded into memory during bootstrap.

The "\*" character must be entered when you specify no value for the memory ("nn") variable, and **do** specify a value for the BIOS source ("z"). In this sort of command line, the "\*" character acts as a space filler so that the computer does not interpret the source value entered as a memory value because it was entered in the memory value space.

During execution, the MOVCPMxx utility will respond with a message in the following form:

```
MOVCPMxx VERSION 2.v
CONSTRUCTING nnk CP/M vers 2.v
READY FOR "SYSGEN" OR
"SAVE 38 CPMnn.COM"
```

## 4 MOVCPMxx EXAMPLES

The following command lines and accompanying explanations are specific examples:

**A>MOVCPM67 64 RETURN**

The system kernel created by this command will be bootable from an H/Z67 drive. It will operate with 64K of RAM. The kernel will be adjusted using the BIOS in computer memory for reference.

**A>MOVCPM37\* C: RETURN**

The system kernel created by this command will be bootable from an H/Z37 drive. It will probe computer memory and operate at computer's memory capacity. This kernel will be adjusted using the BIOS.SYS file stored in drive C: for reference.

**A>MOVCPM1732 B:MYBIOS.SYS RETURN**

This command will adjust a system kernel that will eventually help to make a 5.25-inch hard-sectored disk bootable. It will operate with 32K of RAM. This kernel will be adjusted by referring to the BIOS file "MYBIOS.SYS", which resides in drive B:.

## 5 AFTER RUNNING MOVCPMxx. . .

MOVCPMxx should be immediately followed by the SYSGEN utility, which will transfer the adjusted CP/M system kernel from the TPA to the system tracks of a specified blank disk; or by the SAVE utility, which will transfer the adjusted CP/M system kernel from the TPA to a disk file.

If you perform any other activity immediately after running MOVCPMxx, the work of MOVCPMxx will probably be destroyed.

NOTE: The BSYSGEN utility, although similar to the SYSGEN utility, cannot be used to transfer a system kernel moved by a MOVCPMxx command.

## 6 MOVCPMxx ERROR MESSAGES

### INVALID MEMORY SIZE

EXPLANATION: Valid memory sizes are between 32K and 64K.

### SYNCHRONIZATION ERROR

EXPLANATION: The serial number of the MOVCPMxx utility used must match that of the operating system used.

### READ ERROR

EXPLANATION: MOVCPMxx cannot read data from a file you specified because the file is flawed.

### NO FILE

EXPLANATION: MOVCPMxx cannot read data from a file you specified because it cannot find the file on the specified drive.

### NO SPACE

EXPLANATION: The BIOS file you specified will not fit in the Transient Program Area.

### BAD LOAD

EXPLANATION: A file specified by you did not load properly. You should try the MOVCPMxx command again or specify a different file.

CAN'T OPEN BIOS.SYS

EXPLANATION: MOVCPMxx is unable to use the specified BIOS because it is not stored in a file named "BIOS.SYS", or because it is a defective file.

FATAL ERR F25: NOT ENUF MEMORY

EXPLANATION: MOVCPMxx execution was aborted; you should perform bootstrap and try again.

File not found.

EXPLANATION: MOVCPMxx could not find the file you specified.

# PIP

## *The Utility that Copies Data between Files, Disks, and/or Devices*

PIP stands for Peripheral Interchange Program, the CP/M utility that can be invoked (1, 2, 3) to copy (4) and link (5) files or parts of files (6), and to direct input and output between logical devices (8). PIP also allows files to be transferred between different disks when using a one-drive hardware environment (9). PIP operations can be regulated by parameters (11).

## **1 METHODS OF ENTERING PIP COMMANDS**

Two different methods can be deployed to enter PIP commands: the PIP prompt method and the system prompt method.

## 1.1 PIP Prompt Method

With the PIP prompt mode, you enter “PIP” in response to the system prompt. This entry is sufficient to invoke PIP, which displays its own prompt—the asterisk (\*). You can now enter the argument half of the command line in response to the “\*” prompt supplied by PIP, as shown in the following example:

```
A>PIP RETURN
*{argument} RETURN
*
```

After this PIP operation is finished, PIP again displays the “\*” prompt. In reply, you can enter another argument to command PIP to perform another operation, or enter a carriage return only to return to the operating system and obtain a system prompt.

## 1.2 System Prompt Method

To induce a similar operation without splitting up the command line, the system prompt mode of command entry can be used. With this mode, you must include the argument in the response to the system prompt, as shown in the following example:

```
A>PIP {argument} RETURN
A>
```

After PIP finishes this operation, it automatically returns you to the operating system, and the system prompt is displayed.

All characters specified in such a command line will be automatically translated into UPPER CASE.

NOTE: PIP sends data to a destination from a source, but it does not remove the data from the source. It merely copies an image of the data, and then sends it. Therefore, any data that is copied will remain intact at the source after the PIP operation.



## 2 THE ARGUMENT IN A PIP COMMAND LINE

PIP commands require an argument regardless of which invocation method is used. PIP arguments must specify a “data source” (which can be either a file or a logical input device), and a “data destination” (which can be either a file, a disk, or a logical output device). The “data destination” is entered first and followed by an “=” sign, which is followed by the “data source”, as shown in these examples:

A>**PIP RETURN**  
**\*{data destination} = {data source} RETURN**

OR

A>**PIP {data destination} = {data source} RETURN**

In all cases, copied data is transferred in a right-to-left direction, with respect to the command line components. Hence, the file, disk, or logical output device to receive the data is always specified on the left hand side of the “=” sign; and the file or logical input device to submit the data is always specified on the right hand side of the “=” sign.

If entered in the system prompt mode, characteristics entered in a PIP command line will be in upper case in any future displays. Characters entered in a PIP prompt mode command line will reappear in the case in which they were entered.

### 3 CHARACTERISTICS OF DATA DESTINATION AND SOURCE SPECIFICATIONS

A data destination can be a file, a disk, or a logical device. A data source can be a file or a logical device. Each must be specified in the command line as shown below:

DATA SPECIFICATION CHARACTERISTICS	EXAMPLES
<p>Data can be transferred to a disk by specifying the letter of the drive in which the disk resides, and a colon.</p>	<p>B: = SENDOVER.DOC B: = CON:</p>
<p>Files on non-default disks are identified by specifying the drive in which they reside immediately before the file name.</p>	<p>B:NEW.DOC = C:SEND.DOC B:CREATED.DOC = CON: CON: = B:SENOVER.DOC</p>
<p>Logical devices are specified by entering the three-letter code for that particular device and a colon.</p>	<p>CON: = LST: LST: = SENDOVER.DOC CREATED.DOC = CON:</p>

## 4 FILE COPYING EXAMPLES

The following argument will cause PIP to transfer a copy of the file SENDOVER.DOC from drive B to the same disk, and give it the file name CREATED.DOC. In effect, this operation creates a file backup with a different name on the same disk.

**B:CREATED.DOC = B:SENOVER.DOC RETURN**

The following argument will cause PIP to transfer a copy of the file SENDOVER.DOC from drive C to the disk in drive B, and give it the file name CREATED.DOC.

**B:CREATED.DOC = C:SENOVER.DOC RETURN**

The following argument will cause PIP to transfer a copy of the file SENDOVER.DOC from the currently logged drive to the disk in drive B, and give it the same file name.

**B: = SENOVER.DOC RETURN**

The following argument will cause PIP to transfer a copy of the file SENDOVER.DOC from drive C to the disk in drive B, and give it the same file name.

**B: = C:SENOVER.DOC RETURN**

Each of the following arguments will cause PIP to transfer a copy of the file SENDOVER.DOC from the disk in drive B to the disk in the currently logged drive, and give it the file name SENDOVER.DOC. It should be noted that the data source in this argument is **not** the disk in drive B, but the file SENDOVER.DOC which resides on the disk in drive B. Because the source file is not specified in the typical data source location, PIP assumes that the source file is the same as the destination file, which is specified in its typical location.

**SENOVER.DOC = B: RETURN**

OR

**SENOVER.DOC = B:SENOVER.DOC RETURN**

## 5 DATA SOURCE CONCATENATION

While only one data destination can be specified in a single PIP command line, several data sources can be specified. Thus you can merge data from several locations to one location.

When more than one data source is specified, each data source specification is separated by a comma.

The length of a PIP command line cannot exceed 255 characters. (If the user tries to enter a 256th character into the command line, PIP will begin execution based on the first 255 characters entered.)

PIP will concatenate data sources in the order in which their specifications are entered in the command line.

The following PIP argument will cause PIP to get a copy of the files THISFILE.DOC and THATFILE.DOC from the disk in drive C and the file YOURFILE.DOC from the disk in the default drive. PIP will then combine all three files into one file (in the order they are specified), place this file on the disk in drive B, and assign the name COMBINED.DOC to the transferred combination of the three source files.

**\*B:COMBINED.DOC = C:THISFILE.DOC,C:THATFILE.DOC,YOURFILE.DOC RETURN**

Both files and devices can be specified as data sources in the same command line.

NOTE: If you wish to concatenate object files with a PIP operation, you must specify the O parameter after the file name of each object file that you wish to concatenate.

## 6 COPYING A BLOCK OF DATA

Blocks of files, as well as whole files, can be copied to another file. To do this, you enter a PIP command that specifies the source file name and the beginning and end of the block being transferred. Such an entry requires the use of PIP parameters as in the following example:

```
*BLOCKFIL.TXT = WHOLEFIL.TXT[Sbeginning ^ ZQend ^ Z] RETURN
```

In this example, "BLOCKFIL.TXT" is the name of the destination file to receive a block of text from the source file named "WHOLEFIL.TXT". The block is specified by entering a bracket, the parameter "S", a unique string of text from the "beginning" of the block, a CTRL-Z character, the parameter "Q", a unique string of text from the "end" of the block, a CTRL-Z character, and another bracket.

The following example demonstrates how the user can concatenate blocks from several files, by using the "S" and "Q" parameters to specify the beginning and end of each block:

```
*USBLOCK.TXT = GETADD.TXT[SFour ^ ZQearth ^ Z],CONSTIT.TXT[SWhen ^ ZQAmerica ^ Z] RETURN
```

With the entry of this command line, PIP would combine the specified block from the file "GETADD.TXT" with the specified block from "CONSTIT.TXT". With this data, PIP would create the file "USBLOCK.TXT".

NOTE: The strings following the S and Q parameters will be translated to UPPERCASE if you are using PIP by the System Prompt Method. They will **not** be translated to upper case if you are using the Utility Prompt Method.

## 7 COPYING DATA TO AN INTEL HEXADECIMAL FILE

PIP performs a special function if the data destination is a file with the "HEX" extension (an Intel hex formatted machine code file) and the data source (for one of several data sources) is an input/output device, such as a paper tape reader.

In such a case, PIP checks to ensure that the source is a properly formed Intel hexadecimal file, with legal hexadecimal values and checksum records. If an invalid input record is found, PIP reports an error message at the console and waits for corrective action.

It is usually sufficient to open the reader and rerun a section of the tape (by pulling the tape back about 20 inches). When the tape is ready for the second reading, you should enter a carriage return at the console, and PIP will attempt another read. If the tape position cannot be properly read, then you should simply continue the read (by entering a carriage return after the error message), and enter the record manually, using the ED utility after the disk file is constructed from the other data source components.

For convenience, PIP allows a CTRL-Z ("end-of-file" character) to be entered from the console if the source file is a RDR: device. In such a case, PIP reads the specified device and monitors the keyboard. If a CTRL-Z is entered at the keyboard, then the read operation is terminated normally.

For instance, the following PIP argument could be entered to create a "HEX" file:

**\*CREATED.HEX = CON:,B:SENDOVER.HEX,PTR: RETURN**

The preceding argument will cause PIP to create the data destination file CREATED.HEX by reading the data from source device CON: (as you enter hexadecimal values at the keyboard and eventually enter a CTRL-Z "end-of-file" character), the data from source file SENDOVER.HEX (which includes a CTRL-Z), and finally the data from source device PTR: (until a CTRL-Z is read from the paper tape).

NOTE: Hexadecimal data can be checked for valid format by specifying the "H" or "I" format in the PIP argument.

## 8 COPYING DATA TO OR FROM LOGICAL DEVICES

The following examples, and their ensuing explanations, demonstrate PIP commands that transfer data to and/or from computer-recognized logical devices:

**\*LST: = SENDOVER.PRN RETURN**

The preceding argument will cause PIP to copy the source file SENDOVER.PRN to the LST: device.

**\*CON: = B:THISFILE.ASM,C:THATFILE.ASM,YOURFILE.ASM RETURN**

The preceding argument will cause PIP to concatenate three ASM source files from disk B, disk C, and the logged disk, respectively; and to copy the combination to the CON: destination device.

**\*PUN: = NUL:;SENOVER.ASMmr,EOF:;NUL: RETURN**

The preceding argument will cause PIP to send 40 nulls from the NUL: source device to the PUN: destination device, and then to copy the SENDOVER.ASM source file to the PUN:, then to send a CTRL-Z "end-of-file" character from the EOF: source device to the PUN:, and finally to send 40 more nulls from the NUL: source device to the PUN: data destination device.

## 8.1 Input/Output Devices Accessible through PIP

The PIP utility enables you to transfer data directly to or from the logical and physical devices being used. PIP supports data transfer with respect to the devices indicated in the following table:

LOGICAL DEVICE NAME	PHYSICAL DEVICE NAME	DEVICE DESCRIPTION AND/OR CATALOG NAME OF RECOMMENDED INPUT/OUTPUT MACHINE
CON:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	CRT:	Any non-handshaking RS-232 video terminal at port 0E8H
	BAT:	A batch pseudo device using RDR: for input and LST: for output
	UC1:	Any handshaking RS-232 terminal with ETX/ACK protocol, eg. Diablo KSR 1640 printing terminal
RDR:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	PTR:	Null source, not implemented, returns an "end-of-file" character when accessed
	UR1:	An input modem at port 0D8H
	UR2:	System terminal



PUN:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H
	PTP:	Null sink, not implemented
	UP1:	An output modem at port 0D8H
	UP2:	System terminal
LST:	TTY:	Any non-handshaking RS-232 ASCII terminal at port 0D0H (eg. WH-34)
	CRT:	Any non-handshaking RS-232 video terminal at port 0E8H
	LPT:	line printer (eg. H-14, WH-14, WH-24, H/Z25, WH-36, Epson MX80)
	UL1:	Any Diablo printer

Table 2-3:  
Logical/Physical Devices

You must be certain that the destination device that is specified is capable of receiving data, and that the specified source device is capable of sending data.

### ADDITIONAL LOGICAL DEVICES USED WITH PIP

All of the devices in the preceding table can also be referenced using the STAT utility, which assigns physical devices to logical devices on a temporary basis.

The PIP utility can also gain access to five additional devices, which are defined below:

- NUL: Sends 40 “nulls” (ASCII zeroes) to the device (This device is usually accessed at the beginning and/or end of the output when a paper tape punch device is used.)
- EOF: Sends an “end-of-file” character (ASCII CTRL-Z) to the destination device. (Such a transfer is performed automatically after PIP operation involving a file composed with ASCII characters.)
- INP: Special input source which can be “patched” into the PIP program itself. Through this source, PIP accepts data input character by character using a system call to memory location 103H. The data returns from location 109H. The parity bit must be preset at zero.
- OUT: Special output destination which can be “patched” into the PIP program. PIP transmits data from register C to this destination using a system call to memory location 106H. It should be noted that locations 109H through 1FFH of the PIP memory image are not used, and can be replaced by special purpose drivers using DDT.
- PRN: This device is accessed for the same purposes as the LST: device. In its operation, however, tabs are expanded at every eighth character position, lines are numbered, and page breaks occur every 60 lines. Output directed to the LST: device will be treated identically if the [T8NP] or [T8NP60] parameter is entered with the LST: argument.

## 8.2 Suspending PIP Operations

When sending material with PIP to the CON: logical device, the copy operation can be suspended by entering the *CTRL-S* character at the keyboard, and resumed by entering any character other than CTRL-C. The operation can be aborted by entering any keyboard character other than CTRL-S while data is being transferred. PIP will respond to such an entry with the message:

ABORTED

## 9 USING PIP WITH A ONE-DRIVE HARDWARE ENVIRONMENT

When performing a PIP operation in a one-drive hardware environment, the "PIP" command should be entered alone at the system prompt (using invocation method 1.1) to produce the "\*" "\*" prompt.

When using the PIP utility to copy a file to a file on another disk, the disks involved must be inserted in the drive alternately to allow PIP to read from one file and then write to another. The command line argument should be entered as follows:

**\*B:{destination file name} = A:{source file name} RETURN**

Where "B:" specifies logical drive device B;

where "A:" specifies logical drive device A; and

where both A and B are referred to as physical drive device 0 (the only physical drive device in the hardware environment).

This entry causes PIP to display the message:

Put disk B in drive A: and press RETURN

The disk desired to receive the file copy should be placed in the drive, and a carriage return entered. PIP will then display the following message:

Put disk A in drive A: and press RETURN

The disk containing the data source file should be placed in the drive, and a carriage return entered. You should repeat these two steps alternately, as the PIP prompts indicate, until the entire source file has been copied to the destination file. This process will vary in length, depending on the size of the file being copied. You should be careful to keep track of which disk is A and which is B.

## **10 PIP's METHOD OF OPERATION**

File names and device names can both be used in a single data source argument, with PIP reading each name entered (starting with the name entered on the left-hand side of the source argument) until it reaches "end-of-file". "End-of-file" is indicated by a CTRL-Z character in ASCII files, and by the actual end of the file in non-ASCII files. Each of the data source names entered is read and then concatenated to the preceding name (to its left) until the last name has been read.

The data destination device or file receives a copy of the data from the source files and/or devices. When data from an ASCII file is written, the CTRL-Z "end-of-file" character is appended to the result.

As the PIP operation begins, a temporary file is established in the directory of the destination disk. This file has a name that consists of the primary name specified in the data destination, and a "\$\$\$" extension. This temporary file is not changed to the actual file (with the extension specified in the command line argument) until successful completion of the PIP operation.

Files with a "COM" extension are always assumed to be non-ASCII files.