Chapter 5
Command Directory

This chapter is a directory for the CP/M commands and utility programs contained in the Premium SoftCard IIe System.

Command and Utility Program Guidelines

Commands and utility programs are listed in alphabetical order. In each command and program description, the possible command line formats are shown followed by an explanation of the format. The syntax elements of the format are explained in a "Remarks" section. Where applicable, the different commands that can be used with the utility program are also listed.

This chapter assumes that you know how to use the command or program. If you are unsure of how to use a command or program, see Chapter 6, "CP/M Commands and Utility Programs," in the *Microsoft Premium SoftCard IIe System Installation and Operation Manual.*

APDOS

APDOS [*d*:]*cp/mfilename.ext*=[*s*:]*dosfilename*

Purpose

Copies Apple text and data files from Apple DOS disks to CP/M system disks.

Remarks

d: is the destination disk drive and *s*: is the source disk drive. *cp/mfilename.ext* is the CP/M destination file and *dosfilename* is the Apple DOS source file.

Different procedures are used for copying BASIC files and text files. See "APDOS" in Chapter 6 of the *Microsoft Premium SoftCard IIe System Installation and Operation Manual* for instructions and examples.

ASM

ASM *filename*[*.shp*]

Purpose

Converts a source program written in 8080A assembly language into a HEX file.

Remarks

filename is the name of the source file with an extension of .ASM. The filename extension should not be included in the command line; ASM assumes the file will have an extension of .ASM.

s specifies the disk drive (A: through D:) other than the active drive that contains the source disk.

h specifies the drive that will receive the HEX file. If a HEX file is not needed, *Z* is entered in place of the drive letter.

p specifies which drive should receive the PRN file. A PRN file is the listing of the file with error messages. Enter *Z* to disable the generation of the PRN file. Enter *X* to display the listing on the screen.

If no parameters are specified, ASM assumes that the source file is in the active drive and will create HEX and PRN files as output.

ASM is invoked by typing the ASM command line at CP/M command level. ASM can be stopped or aborted at any time by typing CONTROL-C.

ASM generates two types of error messages. Terminal errors indicate what conditions prevented ASM from assembling the program. Source program errors indicate errors in the source program but don't prevent ASM from assembling the program. All error messages are listed in Appendix A, "CP/M Error Messages."

The following table lists the directives ASM recognizes in addition to the 8080 instruction set.

Table 5.1.

ASM Assembler Directives

Directive	Description
ORG	Define starting address of the program or data section.
END	End program assembly.
EQU	Define a numeric constant.
SET	Set a numeric value.
IF	Begin conditional assembly.
ENDIF	End of conditional assembly.
DB	Define data byte.
DW	Define data word.
DS	Define data storage area.

Examples

```
ASM FONT
```

Assembles the source file FONT.ASM from the active drive. Both the HEX file (FONT.HEX) and the PRN file (FONT.PRN) are saved on the same drive.

```
ASM MACRO.ABX
```

Assembles the source file MACRO.ASM from drive A:. ASM saves the HEX file on drive B: and displays the listing at the terminal.

AUTORUN

AUTORUN [*command line*]

Purpose

Permits you to create startup disks.

Remarks

command line is any executable CP/M program name or CP/M built-in command.

A startup disk must be loaded in the active drive to be executed. The active drive is usually A:. When you start the system, the command line will be executed immediately after the CP/M operating system modules are loaded into memory.

To change the command line on a disk, type *AUTORUN* again with a new command line. Typing *AUTORUN* without a command line deletes the AUTORUN command line from the disk.

Example

AUTORUN CAT

Displays the directory on the default drive when the CP/M is loaded into memory from a cold start.

BOOT

BOOT [{*number*|*M*}]

Purpose

Reboots your Apple IIe computer from any system disk at CP/M command level.

Remarks

number is the slot number (4, 5, or 6) of the disk controller board connected to the disk from which you are loading. If you load the operating system from drive A: or B:, the number can be omitted. (The disk controller board for drives A: and B: is installed in slot 6.)

M allows you to boot from the Apple Monitor in ROM. (The Apple Monitor is the Applesoft™ or Integer BASIC Interpreter in ROM.)

BOOT performs the same function as a CP/M cold start. It can boot Apple DOS, Apple Pascal, Applesoft BASIC, Integer BASIC, or any Apple IIe application software disk.

Make sure that the appropriate disk is in the drive from which you are loading. To load CP/M, type *BOOT* and press the RETURN key. To load any other operating system, type *BOOT* followed by the appropriate argument, and press the RETURN key.

CAT

CAT [*filespec*]

Purpose

Scans the directory of a disk to determine which files are on that disk.

Remarks

filespec is name of the file or files CAT scans for. Wild card characters (? or *) can be used in the filename and extension. CAT (with no arguments) displays an alphabetical list of filenames on a disk in the specified drive.

The list displayed by CAT is in alphabetical order and shows the size of each file and the amount of remaining unused disk space in kilobytes.

Examples

CAT

Scans the disk in the active drive and displays an alphabetical list of files found.

CAT GBASIC.COM

Scans the disk in the active drive for the file GBASIC.COM. If found, it displays the file, the size of the file in kilobytes, and the amount of free storage space remaining on the disk.

COPY

COPY *d*:=*s*:[/*V*]

Copies the contents of one disk onto another.

COPY *d*:/F

Formats a disk.

COPY *d*:/D[/F][/V]

Creates a CP/M data disk.

COPY *d*:/S[/F][/V]

Creates a CP/M system disk.

Purpose

Copies and formats CP/M disks.

Remarks

The *s*: and *d*: arguments indicate the source drive and destination drive. Each of the different functions of COPY are performed by including the software switch in the COPY command line. The software switches and their function are listed in Table 5.2.

Table 5.2.

Software Switches

Switch	Function
/D	Instructs COPY to create a data disk.
/F	Copies the format to disk.
/S	Instructs COPY to only copy the CP/M operating system onto the first three tracks of the disk.
/V	Verifies the copy process.

COPY can be used from either CP/M command level or from COPY program level. COPY is invoked by typing the appropriate command line format and pressing the RETURN key to execute the command.

If you include the /S switch in the COPY command line, COPY will format the disk if it hasn't been formatted previously. If the disk is already formatted, the files on the disk are not deleted. Use the /F switch to delete the previously formatted files.

If the /D switch is used and the disk is already a CP/M system disk, the CP/M system is deleted and an additional 12K bytes of disk space is made available for programs and data.

Important

Avoid using data disks in drive A: and in single-drive systems. The lack of an operating system on data disks prevents CP/M from performing a warm start and recovering from errors.

Examples

```
COPY B:=A:/V
```

Copies the contents of the disk in drive A: onto the disk in drive B: and verifies the copy process by comparing the data contents of the two disks.

```
COPY A:=C:/V/F
```

Formats the disk in drive A: and then copies the contents of the disk in drive C: onto the disk in drive A:.

```
COPY B:/F
```

Formats the disk in drive B:.

```
C:=A:
```

The COPY command line is executed from the program level; it copies the contents of the disk in drive A: onto the disk in drive C:.

```
B:/S
```

The COPY command line is executed from the program level; it copies the operating system software from the disk in drive A: onto the disk in drive B:.

d:

d:

Purpose

Changes the active drive in multiple-drive systems.

Remarks

d: is the disk drive identifier.

DDT

DDT [*filename.ext*]

Purpose

Tests and debugs 8080A assembly language programs.

Remarks

filename.ext is the name of the source file to be examined or modified. The source file must have an extension of .COM or .HEX, or DDT will not recognize it. If you do not enter the filename with the DDT command, DDT is loaded into memory and waits for further instructions.

DDT is the CP/M Dynamic Debugging Tool. It is used in conjunction with the ASM assembler to test and debug assembly programs. You can also use DDT for examining and modifying your programs.

To invoke DDT, type the DDT command line and press RETURN. If you include the filename in the command line, DDT will display the DDT version number, the next available memory location (denoted by NEXT), the program counter setting (denoted by PC), and the DDT program prompt (-). If you enter DDT without the filename, only the version number and the prompt appear.

When the DDT program prompt appears, you can use any of the DDT commands listed in Table 5.3.

Table 5.3.
DDT Commands

Command	Purpose
A*nnnn*	Enters assembly language statements starting at address *nnnn*.
D	Displays the contents of the next 192 bytes of memory.
D*ssss,ffff*	Displays memory contents starting at address *ssss* to address *ffff*.
Fssss,ffff,cc	Fills memory with constant *cc* from address *ssss* to address *ffff*.
G	Begins execution at the address contained in the program counter.
G*ssss*	Begins execution at address *ssss*.
Gssss,bbbb	Sets a breakpoint at address *bbbb*; begins execution at address *ssss*.
G,*bbbb*	Sets a breakpoint at address *bbbb*; begins execution at the address contained in the program counter.
G,*bbbb,cccc*	Sets breakpoints at addresses *bbbb* and *cccc*; begins execution at the address contained in the program counter.
I*filename.ext*	Sets up the default File Control Block using the name *filename.ext*.
L	Lists the next eleven lines of the assembly language program.
L*ssss*	Lists eleven lines of the assembly language program starting at address *ssss*.
L*ssss,ffff*	Lists the assembly language program which starts at address *ssss* and finishes at address *ffff*.

Table 5.3. *(continued)*

Command	Purpose
M*ssss,ffff,dddd*	Moves the memory block (address *ssss* to *ffff*) to address *dddd*.
R	Reads a file from disk.
Rnnnn	Reads a file from disk, beginning at address *nnnn*.
S*ssss*	Displays memory contents at address *ssss* and optionally changes the contents.
T*nnnn*	Traces the execution of *nnnn* program instructions.
U*nnnn*	Executes *nnnn* program instructions, then stops and displays the contents of the CPU registers.
X	Displays the contents of the CPU registers.
X*r*	Displays contents of CPU registers or flag *r* and optionally changes it.

DDT can be aborted at any time by typing CONTROL-C.

Examples

The following example shows how DDT would be invoked and the results of using some of the DDT commands.

```
DDT DUMP.COM
```

Loads DDT and the file DUMP.COM into memory.

```
DDT VERS 2.2
NEXT  PC
1E00  0100
-
```

Displays the DDT version number. NEXT identifies the next free memory location (1E00). PC identifies the program counter setting (0100). "-" is the DDT prompt.

```
L
```

When RETURN is pressed, DDT displays the next 11 lines of assembly language disassembled from memory.

```
0100 LXI   H,0000
0103 DAD   SP
0104 SHLD  0215
0107 LXI   SP,0257
010A CALL  01C1
010D CPI   FF
010F JNZ   011B
0112 LXI   D,01F3
0115 CALL  019C
0118 JMP   0151
011B MVI   A,80
```

DIR

DIR [*d*:][*filename.ext*]

Purpose

Scans a specified disk to determine what files are on that disk.

Remarks

d: is the specified drive and *filename.ext* is name of the file or files DIR scans for. Wild card characters (? or *) can be used in the filename and extension. Entering DIR without any arguments displays only the sequential list of filenames on a disk in the specified drive.

Examples

DIR

Displays all files on the disk in the active drive.

DIR GBASIC.COM

Displays GBASIC.COM on the disk in the active disk.

DIR A:*.COM

Displays all the files with an extension of .COM on disk in drive A:.

DIR B:

Displays all files on the disk in drive B:.

DUMP

DUMP *filespec*

Purpose

Displays the contents of a disk file in hexadecimal form.

Remarks

filespec is the location and name of the file.

To invoke DUMP, type *DUMP* in the command line format at CP/M command level and press RETURN. The hexadecimal contents of the file will be displayed on the terminal's screen. DUMP lists 16 bytes at a time with each line's absolute address on the left.

Example

```
DUMP B:CAT.COM
```

This command line will display the contents of the CAT.COM file in the following format:

```
0000 ED 73 DD 03 31 05 04 CD 45 01 CD 4C 02 0E 11 11
0010 5C 00 CD 05 00 3C 28 16 CD DF 01 CD 80 01 0E 12
...
```

ED

ED *filespec*

Purpose

Creates and edits CP/M ASCII text files.

Remarks

filespec is the location and name of the file to be edited. You must include the extension with the filename. Enter the drive letter (d:) if the file is on a drive other than the active drive.

ED is the CP/M editor. It is used to create and edit CP/M ASCII text files. ED provides the basic requirements for inserting and deleting text, moving from line to line, and searching for text.

At CP/M command level, type *ED* and the filename. Press RETURN to load ED into memory. ED then creates a temporary file (the name of the file with an extension of .$$$) for editing.

When you see the asterisk prompt on the screen, the file is ready to edit. The commands available for editing are listed in Table 5.4.

Table 5.4.
Commands for Editing

Command	Action
*n*A	Moves the number of lines specified by *n* from the temporary file to the edit buffer.
B	Moves the character pointer (CP) to the beginning of edit buffer. The CP takes the place of the cursor.
-B	Moves the CP to end of edit buffer.
*n*C	Moves the CP *n* characters forward.
-*n*C	Moves the CP *n* characters backward.
*n*D	Deletes *n* characters after the CP.
-*n*D	Deletes *n* characters before the CP.
E	Ends edit session, closes files, and returns to CP/M.
*n*F*string* CONTROL-Z	Finds the *n*th occurrence of *string*.
H	Ends edit session, closes and reopens files.
I	Enters insert mode.
I*string* CONTROL-Z	Inserts *string* into the edit buffer.
I*string*	Insert a line of text specified by *string*.
*n*J*fstring*CONTROL-Z*istring1*[,*istring2*, *istring3*,...]CONTROL-Z*estring*CONTROL-Z	The *n* argument specifies how many times the following operation is repeated. Beginning after the CP, ED searches for *fstring*. If found, it inserts *istringn* after it. Then, ED deletes all characters following up to, but not including, *estring*.
*n*K	Deletes *n* lines after the CP.
-*n*K	Deletes *n* lines before the CP.
*n*L	Moves the CP forward *n* lines.
-*n*L	Moves the CP backward -*n* lines.
*n*M*cmdstring* CONTROL-Z	Repeats execution of the ED commands specified by the command string *cmdstring* *n* times.

Table 5.4. *(continued)*

Command	Action
*n*N*string*CONTROL-Z	Searches for the *n*th occurrence of *string* throughout the file.
O	Returns to original file.
*n*P	Moves the CP forward and prints *n* pages.
-*n*P	Moves the CP backward *n* pages and displays the page following the CP.
Q	Quits edit session with no changes saved.
R	Reads temporary file, X$$$$$$$$.LIB into the edit buffer.
R*filename*	Reads library file *filename*.LIB into the edit buffer.
*n*S*fstring*CONTROL-Z*rstring*CONTROL-Z	Searches for *fstring* and replaces with *rstring*. Repeats the operation *n* times.
*n*T	Displays *n* lines preceding the CP.
-*n*T	Displays *n* lines following the CP.
0T	Displays all text from the beginning of the line to the CP.
T	Displays all text from the CP to the end of the line.
0TT	Displays the entire line without moving the CP.
U	Converts text to uppercase.
0V	Displays edit buffer free space in bytes.
V	Verifies line numbers.
*n*W	Writes *n* lines to disk.
*n*X	Copies *n* lines (starting at the CP) to temporary library file X$$$$$$$.LIB.
*n*Z	Delays execution of the command which follows by *n* seconds.
n:	Moves the CP to line number *n*.
[-]*n*	Moves the CP forward or backward and displays one line.

Examples

```
ED DEMO.BAS
```

Loads ED into memory and creates the temporary file DEMO.???. The temporary file is then loaded into the edit buffer.

```
:*
```

ED command prompt; ED is ready for your command.

ERA

ERA *filespec*

Purpose

Erases specified files from a disk.

Remarks

filespec is the location and the name of the file or files to be erased. Wild card characters (? or *) can be used in the filename or extension.

Examples

ERA B:TEMP.OLD

Erases the file TEMP.OLD on the disk in drive B:.

ERA C:*.BAS

Erases all files with the extension .BAS on the disk in drive C:.

ERA *.*

Erases all files on the disk in the active drive.

LOAD

LOAD *filespec*

Purpose

Performs the final step in preparing an assembly language program for execution by converting a disk file with the extension .HEX into a machine-executable command file (with an extension of .COM).

Remarks

filespec is the location and the name of the file with a .HEX extension. The extension need not be included with the filename. LOAD assumes it is a HEX file. Enter the drive letter if the file is on a drive other than the active drive.

At CP/M command level, type *LOAD* in the specified format and press RETURN. LOAD creates a COM file in memory which begins with address 0100H. To save the COM file, use the SAVE command.

Example

```
LOAD B:TIME
```

Loads the TIME.HEX file from drive B:.

```
FIRST ADDRESS      0100
LAST ADDRESS       0222
BYTES READ         0130
RECORDS WRITTEN    02
```

When the file is loaded, the screen displays the starting address (0100), the last address (0222), the number of bytes (130), and the number of records (2) written by LOAD into the file TIME.COM.

MFT

MFT *filespec1*[,*filespec2*...]

Purpose

Copies files from one disk to another on single-drive systems.

Remarks

filespec is the specification of the files to be copied. Wild card characters (? and *) can be used in the file specifications.

MFT is invoked by typing *MFT* at CP/M command level. The copy process is started when you press the RETURN key.

Important

You must have a CP/M system disk in disk drive A: before typing CONTROL-C.

Examples

```
MFT *.COM
```

Copies all COM files on the source disk to the destination disk at CP/M command level.

```
MFT MBASIC.COM,CONFIGIO.BAS
```

Copies the GBASIC.COM and CONFIGIO.BAS files from the source disk to the destination disk at CP/M command level.

PATCH

PATCH {*filespec*|*offset*}=[*p1 p2 p3...*][(*v1 v2 v3*)]

Purpose

Installs program updates and modifications to the CP/M system modules.

Important

The only time you should have to use PATCH is when you receive explicit instructions from Microsoft Corporation. If you wish to install your own modifications or updates without instructions from Microsoft, do so at your own risk.

Remarks

filespec is the name of the COM file to be modified.

offset is a one through six digit hexadecimal byte offset. The offset is from the beginning of the disk if the CP/M system tracks are to be modified.

p1, *p2*, *p3* are two-digit hexadecimal byte "patches."

v1, *v2*, *v3* are optional two-digit hexadecimal verification bytes.

Spaces are required between all byte arguments.

If modifications are made to a COM file, specify the disk and the file by typing the *filespec* argument. If modifications are made to the CP/M system tracks, use the *offset* argument. If the *filespec* is included, the offset is from the beginning of the file starting at byte 0.

The bytes following the equals operator (=) are written to the specified file. If there is no file specified, the bytes are written to the location specified by the offset argument.

Once the patch is made, the asterisk prompt reappears. Repeat the procedure to install another patch, or type CONTROL-C to return to CP/M command level.

PIP

PIP *d*:[*filespec*]=[*s*:]*filespec*[*p*]

Copies a file to another disk.

PIP [*d*:]*newfilespec*=[*s*:]*oldfilespec*[*p*]

Renames the destination file during the copy process.

PIP *d*:[*filespec*]=[*s*:]*filespec*[*gn*]

Copies files from different user areas to the active user area.

PIP [*d*:]*dest*=[*d*:]*source1*,*source 2*...

Appends disk files (concatenation).

PIP LST:=*filespec*[*p*]

Sends data to an output device, such as a printer or terminal.

PIP *ddest*:=*sdest*:[*p*]

Copies data between I/O devices.

Purpose

Copies data between files or devices.

Remarks

d: is the destination drive and *s*: is the source drive.

filespec is the file specification of the file or files from which you are copying. If you are changing the name of the copied file, *newfilespec* is the new filename and *oldfilespec* is the old filename.

[*p*] is the parameter argument. The parameters that can be used with PIP are listed in Table 5.5, "PIP Parameter Summary."

If you are copying files, *dest* is the destination file of the copy operation and *source* is the source file. Commas must separate the source file arguments.

If you are copying data between devices, *ddest*: is the destination device and *sdest:* is the source device of the copy process.

PIP can be used by typing the appropriate command line format. Press RETURN to execute the command. PIP can be aborted at any time by pressing the space bar or any other key during the copying process. PIP confirms that the process has been aborted by displaying the message "ABORTED."

Table 5.5.

PIP Parameter Summary

Parameter	Action
B	Specifies block mode transfer.
D*n*	Deletes all characters after the *n*th column.
E	Echoes the data being copied to the screen during the copy process.
F	Removes formfeed characters from data during the copy process.
G*n*	Copies a file from user area *n* to the active user area.
H	Checks for proper Intel® HEX file format.
I	Ignores any null records in Intel HEX file copy operations.
L	Translates uppercase letters to lowercase.
N	Adds a line number to each line copied.
O	Object file copy operation (ignores end-of-file markers).
P*n*	Inserts page ejects after every *n*th line; the default value is 60 lines.
Q*string*CONTROL-Z	Copies only a portion of the file up to *string*.
R	Directs PIP to copy from a system file.
S*string*CONTROL-Z	Copies only the portion of the file from *string* to the end of the file.
T*n*	Sets tab stops to every *n*th column.
U	Translates lowercase letters to uppercase.
V	Verifies copy by comparison after the copy process has been finished.
W	Directs PIP to copy onto an R/O file.
Z	Zeros the parity bit on input for each ASCII character.

Examples

```
PIP B:=*.BAS
```

Copies all files with the extension of .BAS on the active drive to drive B:.

```
PIP DOG.COM=CAT.COM
```

Copies the file CAT.COM into a new file called DOG.COM on the active drive.

```
B:ED.COM=A:
```

Copies the file ED.COM from drive A: to drive B: under the same name.

```
B:=S*.COM
```

Copies all the files on the active drive that start with the letter "S," and have an extension of .COM to drive B:.

REN

REN [*d*:]*new filename.ext=old filename.ext*

Purpose

Renames files while leaving the file text intact.

Remarks

new filename.ext is the new name of the file and *old filename.ext* is the original name of the file. Wild card characters cannot be used in either the old filename or the new filename.

Examples

```
REN TEMP.NEW=TEMP.OLD
```

Renames TEMP.OLD as TEMP.NEW.

```
REN B:PEAR.COM=APPLE.COM
```

Renames APPLE.COM on drive B: as PEAR.COM.

SAVE

SAVE *nnnfilespec*

Purpose

Saves the contents of memory in a specified disk file.

Remarks

nnn is the number of memory pages to be saved.

filespec is the drive and the name of the file in which to save the memory contents.

Example

```
SAVE 26 C:MYPROG.COM
```

Saves 26 pages of memory in a file called MYPROG.COM on disk drive C:.

STAT

STAT [*d*:]

Displays disk drive status.

STAT *d*:{DSK:|USR:}

Displays active disk and user area status.

STAT *filespec*

Displays file status.

STAT {*d*:|*filename.ext*}$*attribute*

Assigns attributes to files and disks.

STAT *log:=phy*:

Makes device assignments.

STAT VAL:

Displays possible STAT commands.

STAT DEV:

Displays the current device assignments.

Purpose

Displays status information and changes device assignments.

Remarks

d: is the disk drive identifier.

filespec is the name of the file or files from which you want to obtain status information. Wild card characters can be used to obtain status information on more than one file at a time.

attribute is one of the attributes from Table 5.6, "File and Disk Attributes," that can be assigned to the file or disk.

log: and *phy*: are the logical and physical I/O devices.

STAT is executed by typing the appropriate command and pressing the RETURN key. STAT is executed from CP/M command level only.

Table 5.6.

File and Disk Attributes

Attribute	Action
$R/O	Prevents writing to or deleting the file.
$R/W	Allows writing to and deleting the file. This attribute cancels $R/O.
$SYS	Prevents the display of the file when the DIR built-in command is invoked.
$DIR	Cancels the $SYS attribute.

Examples

STAT

Displays file attributes and amount of free space (in kilobytes) for all disk drives since the last warm or cold start.

STAT B:

Displays amount of disk free space in drive B:.

STAT DEMO.BAS

Displays size and attributes of DEMO.BAS file on the active drive.

STAT B:DOG.COM $R/O

Assigns the $R/O attribute to DOG.COM on drive B:.

STAT CON:=TTY:

Assigns the physical device TTY: to the logical device CON:.

STAT C:$R/O

Assigns a temporary write-protect status to drive C:.

SUBMIT

SUBMIT *filespec abc*

Purpose

Creates a file which contains commands to be executed from a disk file rather than from the keyboard.

Remarks

filespec is the location and filename of a text file to be submitted. The filename must have a .SUB extension. The extension need not be included with the filename; SUBMIT assumes it is a SUB file. Enter the drive letter if the file is on a drive other than the active drive.

a, *b*, and *c* are arguments for optional variables in the SUBMIT file. The variables can be filenames or other information needed by the commands in the SUBMIT file. The symbols $1, $2, and $3 are substituted for missing parameters in format 2.

Examples

File: TEST.SUB

The name of the SUBMIT file.

CAT $1.BAS

The contents of the SUBMIT file.

```
PIP $2:=$1.BAS
GBASIC $1
SYSTEM
ERA $1.BAS
```

This program looks for a GBASIC file named by variable $1. PIP copies the file to the drive named by variable $2. GBASIC then executes the file. The file is erased after execution.

SUBMIT TEST DEMO B

Loads SUBMIT into memory and creates the SUBMIT file, $$$.SUB, from the file, TEST.SUB. $$$.SUB executes the commands from TEST.SUB: and searches for DEMO.BAS. The SUBMIT command then copies it to drive B:, runs GBASIC and DEMO.BAS, and erases DEMO.BAS after execution.

TYPE

TYPE *filespec*

Purpose

Displays the contents of a specified text file on the screen.

Remarks

filespec is the location and the name of the file. No wild card characters are allowed in the filespec.

Example

```
TYPE DUMP.ASM
```

Displays the contents of the file DUMP.ASM on the screen.

USER

USER *n*

Purpose

Separates disk memory into user areas.

Remarks

The user areas are designated by numbers. *n* is the number of the user area.

XSUB

XSUB

Purpose

XSUB is a variation of SUBMIT, which allows constant character input from a disk file during program execution.

Remarks

Introduce XSUB as the first line of a SUBMIT file (*filename.SUB*). Run the SUBMIT file as instructed by the command prompts. When CP/M processes the SUBMIT file, it relocates the XSUB program directly below the CCP in memory in order to process the command lines of the SUBMIT file. The XSUB program remains active until all the commands in the SUBMIT file have been executed or until a cold start has been performed.