Appendix A
CP/M Error Messages

This appendix lists in alphabetical order the error messages for the SoftCard implementation of CP/M. Each error message includes the possible causes and what actions you may take in response to them. In each of the error message descriptions, *d*: represents the disk drive identifier (A:-D:).

Aborted

Cause. PIP stopped; a key was pressed by the user.

Action. Retry the command.

Bad Delimiter

Cause. The wrong delimiting character was used in the STAT command line.

Action. Check for typing errors and try the command again.

BDOS ERR ON *d*: Bad Sector

Cause. An attempt was made to execute a command (built-in or transient) when:

There was no disk in the specified drive

The drive door was not closed

The disk was inserted into the specified drive improperly

The drive was not connected to a disk controller board (see SELECT error)

The disk was damaged or worn

Action. Correct the error condition. Then press the CONTROL-C keys to perform a warm start. Retry the command.

BDOS ERR ON *d*: File R/O

Cause. A write operation was attempted to a file that has a Read Only (R/O) attribute.

Action. Type any character to perform a warm start and return to CP/M command level.

BDOS ERR ON *d*: R/O

Cause. The disk in the accessed drive was changed without pressing CONTROL-C; or there is a write-protect tab on the disk.

Action. Press any key to perform a warm start and return to CP/M command level.

BDOS ERR ON *d*: Select

Cause. An attempt was made to access a disk drive when either the drive was not connected to a controller, or the controller has been installed in the wrong accessory slot.

Note that if you have only one drive attached to a disk controller board, an attempt to access a drive that is not installed results in a BAD SECTOR error instead of a SELECT error.

Action. Press any key to perform a warm start and return to CP/M command level.

Cannot Close Destination File-*filespec*

Cause. The output file specified in the PIP command line cannot be closed. This is usually caused by a write-protect tab on the disk.

Action. Remove the write-protect tab from the disk and try the command again.

Cannot Close File

Cause. A write operation has been attempted to a disk that has a write-protect tab on it.

Action. Make sure the disk is not write-protected and try the command again.

Cannot Read

Cause. The PIP program cannot read the source device specified in the command line.

Action. Check the RDR: device assignment and the physical connections to the current reader device.

Cannot Write

Cause. An invalid destination device was specified in the PIP command line.

Action. Check the device assignments and retry the command.

Checksum Error

Cause. PIP encountered a hex checksum record error while copying a hex file.

Action. Recreate the hex file with an assembler program and retry the command.

Checksum Error Load Address ...

Cause. The file specified in the LOAD command line contains errors.

Action. Recreate the hex file with an assembler program and retry the command.

Command Buffer Overflow

Cause. There are too many characters in the SUBMIT command buffer.

Action. Make sure that the submit input file doesn't exceed 2048 characters.

Command Error

Cause. Either there is a syntax error in the command line or the command is not understood (i.e., the arguments in the command line were not recognized by the program). Command Error is generated by utility programs written by Microsoft.

Action. Retype the command line in the correct format and retry the command.

Command Too Long

Cause. A command in the submit input file is longer than 125 characters.

Action. Check the commands in the submit input file and re-submit the input file.

Correct Error, Type Return or CTRL-Z

Cause. A hex record checksum error was encountered during the transfer of a hex file by PIP.

Action. Correct the hex file and retry the command.

Destination is R/O, Delete (y/n)

Cause. The destination file specified in the PIP command line is designated R/O.

Action. Enter *Y* to delete the existing file and PIP will complete the copy process. Enter *N* to abort the copy process.

Directory Full

Cause. An attempt was made to copy files to a destination disk which has no more storage space.

Action. Insert another disk in the destination drive and retry the command.

Disk Full

Cause. An attempt was made to copy files to a destination disk which has no more storage space. This error message is generated by the APDOS, MFT, or UPLOAD programs.

Action. Insert another disk in the destination drive and retry the command.

Disk I/O Error

Cause. The COPY program cannot format the disk. This is caused by either a bad or a worn-out disk, or the disk drive door is not closed.

Action. Ensure that the disk drive door is closed. If the same error message appears, replace the disk.

Disk Read Error

Cause. The source file specified in the PIP command contains an end-of-file character in the wrong place.

Action. Make sure that the end-of-file character is in the right place.

Disk Write Error

Cause. A disk write operation was attempted to a full disk.

Action. Either erase some files or try the write operation with another disk.

Disk Write-Protected

Cause. An attempt was made to write to a disk that has a write-protect tab on it. This error message is generated by the APDOS and COPY programs.

Action. Remove the write-protect tab from the disk and retry the command.

Error:Bad Parameter

Cause. There is an illegal parameter in the PIP command line.

Action. Check the command line and retry the command.

Error:Cannot Close File, Load Address *xxxx*

Cause. An error exists in the program being loaded with the LOAD program. The disk may be write-protected.

Action. Check the source program for errors. Check the disk for a write-protect tab.

Error:Cannot Open Source, Load Address *xxxx*

Cause. The LOAD program cannot find the file specified in the LOAD command line; or no filename was specified.

Action. Check the filename of the source file to be loaded. Make sure the filename is included in the LOAD command line. Retry the command.

Error:Disk Inverted, Load Address *xxxx*

Cause. The address of a record was too far from the address of the previously processed record.

Action. Use DDT to read the hex file into memory, then use the SAVE command to store the file back to the disk. Retry the LOAD command.

Error:Disk No More Directory Space, Load Address *xxxx*

Cause. The destination disk in the active drive is full.

Action. Change disks and retry the command.

Error:Disk Read, Load Address *xxxx*

Cause. The file specified in the LOAD command line cannot be found on the disk.

Action. Check to see that the file exists on the disk in the active drive.

Error:Disk Write, Load Address *xxxx*

Cause. The destination disk in the active drive is full.

Action. Change disks and retry the command.

Error On Line *nnn*

Cause. There is an error in the submit input file at the specified line number (*nnn*).

Action. Correct the error and retry the command.

File Error

Cause. The disk is full and the ED program cannot write any more data to the accessed file.

Action. Copy the file to another disk or delete other files from the same disk.

File Exists

Cause. An attempt was made to change the name of a file to an existing filename.

Action. Make sure the "new" filespec argument in the REN command line does not match any existing filenames on the same disk. Retry the command.

File Exists, Erase It

Cause. The destination file named in the ED command line already exists.

Action. Place the destination file on another disk or in a different user area.

File Is Read Only

Cause. The file specified in the ED command line has an R/O attribute.

Action. Change the file status with the STAT program.

File Not Found

Cause. The source file specified in the APDOS, AUTORUN, CAT, COPY, MFT, PATCH, STAT, or UPLOAD command line does not exist.

Action. Check the spelling of the filename and reenter the command line.

Filename?

Cause. An incorrect use of wild card characters in the REN command line.

Action. Retry the command with no wild card characters in the command line.

Filename Required

Cause. ED was invoked without a filename argument in the command line.

Action. Include a filename in the ED command line.

hhhh?? = dd

Cause. The mnemonic (*dd*) at address (*hhhh??*) is not an 8080 or Z80 assembly language instruction.

Action. Correct the mnemonic.

Insufficient Memory

Cause. There is not enough memory available to load the specified file with the DDT R or E command.

Action. Reduce the size of the file and load in segments of the file.

Invalid Assignment

Cause. One of the device names specified in the STAT command line is either misspelled or cannot be assigned to the other specified device.

Action. Check the spelling of the device name and retry the command. If the same error message appears again, check for the valid device assignments by typing *STAT VAL:*.

Invalid Control Character

Cause. An invalid CONTROL character was included in a submit input file.

Action. Use only ^A through ^Z CONTROL characters in a submit input file.

Invalid Digit

Cause. The hex file specified in the PIP command line contains an invalid hex digit.

Action. Correct the hex file and retry the PIP command.

Invalid Disk Assignment

Cause. An attempt was made to assign an attribute other than R/O to a disk drive with the STAT program.

Action. Assign only the R/O attribute to disk drives. Remove the R/O attribute with the STAT program.

Invalid Disk Select

Cause. A command line specified a nonexistent disk drive.

Action. Specify only disk drives A: through D: in the command line. Check for any loose connections to the disk drives and for unformatted disks.

Invalid File Indicator

Cause. STAT did not recognize the attribute in the STAT command line.

Action. Specify only R/O, R/W, DIR, or SYS in the STAT command line.

Invalid Format

Cause. The PIP command line was in the wrong format.

Action. Check the command line format and retry the command.

Invalid Hex Digit ...

Cause. The file specified in the LOAD command line contains an incorrect hex digit.

Action. Correct the file and retry the command.

Invalid Separator

Cause. An invalid separator character was used in the PIP command line.

Action. Check the command line format and retry the command.

Invalid User Number

Cause. An invalid user number was specified in the PIP command line.

Action. Use only user numbers 1 through 15.

n?

Cause. A number greater than 15 was specified in the USER command line.

Action. Use only user numbers 1 through 15.

No Directory Space

Cause. There is no room on the disk for the .PRN and .HEX files generated by ASM.

Action. Either delete files from the active drive or specify another drive.

No File *filespec*

Cause. The file specified in the command line cannot be found.

Action. Recheck the spelling of the filespec and try again.

No File(s) Found, *xxx*k Bytes Available

Cause. The file specified in the CAT command line does not exist.

Action. Check the spelling of the filename and reenter the command line.

No Input File Present On Disk

Cause. The file specified in the DUMP command line does not exist.

Action. Recheck the spelling of the filespec and try again.

No Source File Present

Cause. The ASM assembler could not find the file specified in the command line.

Action. Check spelling of the file and ensure that the disk is listed in the disk directory. Retry the command.

No Space

Cause. An attempt was made to save the contents of memory with the SAVE command, but there is no space left on the disk.

Action. Use a disk with sufficient storage space and retry the command.

No Sub File Present

Cause. The SUBMIT program was run but no submit input file was specified.

Action. Create a submit input file.

Not A Character Source

Cause. An invalid source device was specified in the PIP command line.

Action. Use either RDR: or CON: as source devices.

Not Deleted

Cause. The file specified in the PIP command line has an R/O attribute and cannot be deleted.

Action. Change the status of the file with the STAT program.

Not Found

Cause. PIP cannot find the file specified in the command line.

Action. Check the spelling of the file and try again.

Output File Write Error

Cause. Either a file with write-protect status has been specified as the ASM destination file, or there is no free space left on the disk.

Action. Check the attributes of the destination file and the amount of free disk space with the STAT program.

Parameter Error

Cause. The submit input file contains an invalid parameter.

Action. Use only valid parameters ($0 through $9) in the submit input file.

Quit Not Found

Cause. The *Q* parameter was specified in the PIP command line but there is no string argument in the input file.

Action. Insert the appropriate string argument in the input file specified by PIP.

Read Error

Cause. The file specified in the TYPE command line contains an error.

Action. Use the STAT program to check the disk and the file. Retry the command.

Reader Stopping

Cause. The read operation from the RDR: device has been interrupted. (A key was pressed during the read operation.)

Action. Retry the command.

Record Too Long

Cause. The file or device specified in the PIP command line contains a record longer than 128 bytes.

Action. Use the STAT program to check for any records longer than 128 bytes.

Source File Name Error

Cause. Wild card characters * and ? were specified in the source filename argument of the ASM command line.

Action. Specify only one source filename in the ASM command line.

Source File Read Error

Cause. The file read by the ASM assembler is in the wrong format or has instructions the ASM assembler cannot understand.

Action. Check the file for the proper format and check that the assembly language instructions are 8080 mnemonics.

Start Not Found

Cause. PIP cannot find the string argument in the input file specified by the S parameter.

Action. Check the input file for the appropriate string argument.

Too Many Files

Cause. STAT cannot process the files specified. Either there are too many files (more than 512), or there is not enough free RAM available to process the files.

Action. Delete or transfer files to another disk. Retry the command and specify fewer files.

Unexpected End Of Hex File *filespec*

Cause. The hex file specified in the PIP command line contains an end-of-file character before a termination hex record.

Action. Correct the hex file and retry the command.

Unrecognized Destination

Cause. PIP did not recognize the destination file or device specified in the command line.

Action. Make sure that the destination device is a currently assigned device, or that the destination file exists.

Use: STAT *d*:=R/O

Cause. The drive argument (*d*:) in the STAT command line was used in the wrong format.

Action. Use the proper format (STAT *d*:=R/O) for the drive argument.

Verify Error

Cause. The data copied onto a destination disk does not match the data on the source disk. This is caused by a worn or damaged disk, or by hardware problems. The VERIFY ERROR message is generated by the COPY and PATCH transient programs.

Action. Try using a different disk and repeat the command. If the same error message appears again, check the connections to the disk drives and the disk controllers. If there is a hardware problem, contact your dealer.

? (Syntax error)

Cause. The command was not understood. Either the command was mistyped, or invalid arguments were included in the command line.

Action. Retype the command line in the correct format.

Appendix B
Downloading to the SoftCard

The Premium SoftCard IIe System allows you to copy CP/M files from CP/M-based computer systems to Apple systems through an RS-232 Serial I/O port. UPLOAD and DOWNLOAD programs permit you to copy programs that are in a different disk format (for example, on 8-inch disk drives) to the SoftCard CP/M format (5-1/4-inch disk drives).

The UPLOAD program is a source listing that must be entered into the source computer (a system that does not support SoftCard) as part of the communication link. UPLOAD must be configured for the source computer's specific I/O environment with the DDT transient program.

DOWNLOAD is the counterpart to the UPLOAD program that must be entered into your Apple IIe computer system.

Warning

Use of UPLOAD and DOWNLOAD assumes familiarity with 8080 assembly language programming. These programs are intended for experienced programmers only!

Program Requirements

To use the UPLOAD and DOWNLOAD programs, you will need the following:

1. A working knowledge of the DDT transient program and 8080 assembly language programming.

2. A CP/M-based computer system (in addition to your Apple IIe) with an RS-232 Serial I/O port other than the port used for console I/O.

3. An Apple Communications Interface board or a CCS 7710A Serial Interface board installed in slot 2 of the Apple IIe computer.

4. An RS-232 Serial I/O port cable.

Downloading Procedure

The following procedure shows how to copy files from another CP/M system to your Apple IIe system.

1. Insert a copy of the SoftCard Master disk into drive A:.

2. Load CP/M with a cold start and then load the DDT transient program.

3. Using DDT, enter the following machine-language program (UPLOAD) into the source computer starting at address 0163H:

```
0163 3A 80 00 B7 11 D7 01 CA CC 01 CD 03 01
0170 0E 0F 11 5C 00 CD 05 00 3C 11 E5 01 CA CC 01 3E
0180 52 CD 43 01 CD 23 01 FE 53 C2 7F 01 3E 47 CD 43
0190 01 11 06 02 CD D2 01 0E 14 11 5C 00 CD 05 00 B7
01A0 C2 C9 01 21 80 00 0E 00 16 80 7E CD 43 01 A9 4F
01B0 23 15 C2 AA 01 79 CD 43 01 CD 23 01 FE 42 CA A3
01C0 01 FE 47 CA 97 01 C3 B9 01 11 F4 01 CD D2 01 C3
01D0 00 00 0E 09 C3 05 00 43 6F 6D 6D 61 6E 64 20 45
01E0 72 72 6F 72 24 46 69 6C 65 20 6E 6F 74 20 66 6F
01F0 75 6E 64 24 0D 0A 55 50 4C 4F 41 44 20 43 6F 6D
0200 70 6C 65 74 65 24 55 70 6C 6F 61 64 69 6E 67 2E
0210 2E 2E 24 FE
```

4. Enter the following three bytes (a JMP 0163H instruction) at address 0100H:

   ```
   C3 63 01
   ```

   When you have finished, check the data entered. Use the DDT "L" command to list the program and compare it with the displayed UPLOAD listing in this section. Before making any of the following program modifications, exit DDT and save the program by typing in

   ```
   SAVE 2 UPLOAD.COM.
   ```

   and pressing the RETURN key.

5. Modify the UPLOAD program to recognize the RS-232 Serial I/O port on the source computer. Use DDT to write subroutines at the three memory addresses outlined in the following list. Each subroutine must begin at the address specified next to the subroutine. 32 bytes are allocated for each subroutine.

Subroutine	*Address*
INITSER	0103H

This subroutine initializes the serial port on the source machine (baud rate, data format, etc.). The data format should be set up for eight data bits, one stop bit, no parity, and for compatibility with the Apple Communications Interface board or CCS 7710A Serial Interface board.

Subroutine	*Address*
RETSER	0123H

If no character is available at the serial port, this subroutine must return A=00. If a byte is available, the routine should read the byte and return it in the A register.

Subroutine	*Address*
WRSER	0143H

This subroutine outputs a byte to the serial port. You must save all registers including register A.

Once the routines are written into the UPLOAD program, save the program in a disk file with the SAVE command.

6. Connect an RS-232 Serial I/O port cable from the Apple IIe to the source computer. One port must be configured as a send (DTE) device, and the other as a receiver (DCE). The XMIT and RCVE lines (pins 2 and 3) may have to be reversed, or certain "handshaking" lines connected. If you are using a CCS 7710A Serial Interface board, connect pins 4, 6, and 20 together on the Apple port. Ensure that the data formats used by the two serial ports are the same.

7. Once UPLOAD has been loaded into the source computer and the connecting cable is configured properly, you are ready to start transferring data.

   On the Apple IIe, type

   ```
   DOWNLOAD filespec
   ```

   filespec is the name of the file to be transferred.

   On the source computer, type

   ```
   UPLOAD filespec
   ```

   As soon as communication is established, the Apple displays

   ```
   DOWNLOADING
   ```

   and the source computer displays

   ```
   UPLOADING...
   ```

   As each 128-byte record is transferred successfully, a period (.) is displayed on the Apple screen. If an error is detected during the transfer of a 128-byte record, a "B" is printed, and the record is retransmitted.

When the transfer process has been completed, the source computer displays

```
UPLOAD COMPLETE
```

and returns to CP/M command level. When this message appears on the source computer, press CONTROL-C. The Apple then displays

```
DOWNLOAD COMPLETE
```

and returns to CP/M command level.

8. This step completes the data transfer procedure. Step 7 must be repeated for each file to be transferred. To transfer more than one file at a time, use the SUBMIT transient program to create a batch file. The DOWNLOAD and UPLOAD programs can also be modified for use with serial communication accessory boards other than those listed in "Program Requirements" at the beginning of this appendix. The source listing for UPLOAD and DOWNLOAD is provided for this purpose.

UPLOAD Source Listing

```
                        ;
                        ;                UPLOAD
                        ;          (C) 1980 MICROSOFT
                        ;
0000    =       BOOT    EQU     0000H   ;Boot system
0005    =       BDOS    EQU     0005H   ;BDOS entry point
005C    =       FCB     EQU     005CH   ;Default FCB
0080    =       BUFFER  EQU     0080H   ;Default buffer address
0100            ;       ORG     0100H   ;Start at TPA
                        ;
0100    C36301  UPLOAD  JMP     ENTRY   ;Jump around these
                        ;
                        ;
                INIT:                   ;Initialize serial port
                        ;
                ; This subroutine should perform any required initialization of the serial
                ; port. If none is required, a 'RET' will do.
                        ;
0103                    DS      32      ;Space for routine
                INPTS:                  ;Input status/read
                        ;
                ; The input status/read routine returns zero in register A if no byte is
                ; available. If a byte is available, the byte should be read and returned in
                ; the A register.
                        ;
0123                    DS      32      ;Space for routine
                OUTPUT:                 ;Send a byte to Apple
                        ;
                ; The output routine should transmit the byte found in the A register to
                ; the Apple through the serial port. All registers including A should be
                ; saved.
                        ;
0143                    DS      32      ;Space for routine
                        ;
                        ;
0163    3A8000  ENTRY   LDA     Buffer  ;Make sure filename is typed
                        ;
0166    B7              ORA     A       ;Is there a non-zero number of
                                        ;characters in the command line?
                        ;
0167    11D701          LXI     D,CMDMSG ;Default to command error
                                        ;message
016A    CACC01          JZ      EXIT    ;Quit
016D    CD0301          CALL    INIT    ;Initialize serial port
0170    0E0F            MVI     C,15    ;Open file command
0172    115C00          LXI     D,FCB   ;Point to FCB
0175    CD0500          CALL    BDOS    ;Open it up
0178    3C              INR     A       ;FF becomes zero
0179    11E501          LXI     D,FNFMSG ;Default to file not found message
                        ;
017C    CACC01          JZ      EXIT    ;No file
                        ;
                ; Send 'R' characters until DOWNLOAD answers
                        ;
```

```
017F  3E52      RDYLP   MVI   A,'R'          ;Send 'R' for 'READY'
0181  CD4301            CALL  OUTPUT         ;Send via serial line
0184  CD2301            CALL  INPSTS         ;Response received?
0187  FE53              CPI   'S'            ;'S' for 'SET'
0189  C27F01            JNZ   RDYLP          ;Then try again
018C  3E47              MVI   A,'G'
018E  CD4301            CALL  OUTPUT         ;Send to DOWNLOAD
0191  110602            LXI   D,WRKMSG       ;Indicate downloading is in
                                             ;progress
                                             ;
0194  CDD201            CALL  PRMSG
0197  0E14      READ    MVI   C,20           ;Read sequential function
0199  115C00            LXI   D,FCB
019C  CD0500            CALL  BDOS           ;Go read 128 bytes
                                             ;
019F  B7                ORA   A              ;Error?
01A0  C2C901    JNZ     EOF                  ;End of file
01A3  218000    TRYAGN: LXI   H,BUFFER       ;Point to the 128 bytes
                                             ;
01A6  0300              MVI   C,0            ;Checksum = 0
01A8  1680              MVI   D,80H          ;Byte count = 128
                                             ;
01AA  7E        LOOP1   MOV   A,M            ;Get character
01AB  CD4301            CALL  OUTPUT         ;Send it (must save A)
01AE  A9                XRA   C              ;Calculate checksum
01AF  4F                MOV   C,A            ;Update it
01B0  23                INX   H              ;Point to the next byte
01B1  15                DCR   D              ;Decrement byte count
01B2  C2AA01            JNZ   LOOP1          ;Keep going until all 128 are sent
01B5  79                MOV   A,C            ;Now send checksum byte
01B6  CD4301            CALL  OUTPUT         ;Send it
                                             ;
                ;Wait for verification: 'G'=GOOD, 'B'=BAD
                                             ;
01B9  CD2301    VFYLP:  CALL  INPSTS         ;Get a character
01BC  FE42              CPI   'B'            ;A bad read?
01BE  CAA301            JZ    TRYAGN         ;Start again
01C1  FE47              CPI   'G'            ;A good read?
01C3  CA9701            JZ    READ           ;Go get next record then
01C6  C3B901            JMP   VFYLP          ;Character must not have been
                                             ;ready
                                             ;
01C9  11F401    EOF:    LXI   D,DONMSG
01CC  CDD201    EXIT:   CALL  PRMSG          ;Output message
01CF  C30000            JMP   BOOT           ;All finished
                                             ;
01D2  0E09      PRMSG:  MVI   C,9            ;Print message function
01D4  C30500            JMP   BDOS
                                             ;
                                             ;
01D7  436F6D6D61CMDMSG:   DB  'COMMAND ERROR$'
01E5  46696C6520FNFMSG:   DB  'FILE NOT FOUND$'
01F4  0D0A55504CDONMSG:   DB  13,10,'UNLOAD COMPLETE$'
0206  55706C6F61WRKMSG:   DB  'UNLOADING....$'
                                             ;
0213            END
```

DOWNLOAD Source Listing

```
                        ;
                        ;               DOWNLOAD
                        ;
                        ;This program works with an Apple Communications Interface or a CCS
                        ;7720A Serial Interface in slot two.
                        ;
                        ;
                        ;
                        ;               (C) 1980 MICROSOFT
                        ;
0000    =       BOOT    EQU     0000H   ;Boot system
0005    =       BDOS    EQU     0005H   ;BDOS entry point
005C    =       FCB     EQU     005CH   ;Default FCB
0080    =       BUFFER  EQU     0080H   ;Default buffer address
                        ;
E0AE    =       COMSTS  EQU     0E0AEH  ;Communication card status
                                        ;location
E0AF    =       COMDAT  EQU     0E0AFH  ;Communication card data—slot 2
E000    =       APPKBD  EQU     0E000H  ;Apple keyboard
                        ;
                        ;
0100                    ORG     0100H   ;Start at TPA
                        ;
                        ;
                        ;
0100    3A8000  DWNLOD: LDA     BUFFER  ;Make sure there's a filename
0103    B7              ORA     A       ;Any characters in command line?
0104    11C001          LXI     D,CMDMSG ;Point to CMD error message
0107    CA8D01          JZ      EXIT    ;Quit
010A    0E13            MVI     C,19    ;Delete file
010C    115C00          LXI     D,FCB
010F    D5              PUSH    D       ;Save PTR: to FCB
0110    CD0500          CALL    BDOS
0113    D1              POP     D       ;Reget PTR: to FCB
0114    0E16            MVI     C,22    ;Make file
0116    CD0500          CALL    BDOS
                        ;
0119    3C              INR     A       ;Check for error
011A    11CE01          LXI     D,NDSMSG ;Get ready to print
                                        ;'NO DIR. SPACE'
011D    CA8D01          JZ      EXIT
                        ;
                ;Wait till UPLOAD sends an 'R'
                        ;
0120    CDA001  RDYLP:  CALL    RDCOM   ;Get a character from
                                        ;communication card
0123    FE52            CPI     'R'     ;'R' for 'READY'?
0125    C22001          JNZ     RDYLP
                        ;
0128    1E53            MVI     E,'S'   ;Get 'S' for 'SET'
012A    CD9301          CALL    WRCOM
                        ;
```

```
012D CDA001   GETGEE:  CALL   RDCOM      ;Wait for 'G' for 'GO'
0130 FE47              CPI    'G'
0132 C22D01            JNZ    GETGEE
                                         ;
0135 21F501            LXI    H,WRKMSG   ;Point to 'DOWNLDNG' message
0138 7E        PRLP:   MOV    A,M        ;Get character
0139 B7                ORA    A          ;Set CC's
013A CA4701            JZ     TRYAGN     ;Go do DOWNLOAD
013D E5                PUSH   H
013E 5F                MOV    E,A        ;Character to register E for
                                         ;CONOUT
013F CDBB01            CALL   CONOUT
0142 E1                POP    H
0143 23                INX    H
0144 C33801            JMP    PRLP
                                         ;
                                         ;
0147 218000    TRYAGN: LXI    H,BUFFER   ;Point to 128-byte buffer
014A 0E00              MVI    C,0        ;Clear checksum
014C 0E81              MVI    C,81H      ;Read 128 bytes + 1 checksum
                                         ;
014E CDA001    LOOP1   CALL   RDCOM      ;Read a byte
0151 77                MOV    M,A        ;Store it
0152 A9                XRA    C          ;Calculate checksum
0153 4F                MOV    C,A        ;Update it
0154 23                INX    H          ;Next byte
0155 15                DCR    D          ;Decrement byte count
0156 C24E01            JNZ    LOOP1      ;Not done--continue
0159 B7                ORA    A          ;Was checksum zero?
015A CA6A01            JZ     GOODRD     ;Things are OK
                                         ;
015D 1E42      BADRD:  MVI    E,'B'      ;'B' for 'BAD'
015F CDBB01            CALL   CONOUT
0162 1E42              MVI    E,'B'      ;Send 'B' to upload
0164 CD9301            CALL   WRCOM
0167 C34701            JMP    TRYAGN
                                         ;
016A 1E2E      GOODRD: MVI    E,'.'      ;Print a period
016C CDBB01            CALL   CONOUT
016F 115C00            LXI    D,FCB      ;Point to FCB
0172 0E15              MVI    C,21       ;Write sequentially
0174 CD0500            CALL   BDOS
0177 1E47              MVI    E,'G'      ;Send UPLOAD 'G' for 'GOOD'
0179 CD9301            CALL   WRCOM
017C C34701            JMP    TRYAGN
                                         ;
                                         ;
017F 3210E0    DONE:   STA    APPKBD+10H ;Clear keyboard strobe
0182 115C00            LXI    D,FCB
0185 0E10              MVI    C,16       ;Close the file
0187 CD0500            CALL   BDOS
018A 11E101            LXI    D,DONMSG   ;All done message
```

```
018D CDB601   EXIT:    CALL   PRMSG     ;Print the message
0190 C30000            JMP    BOOT
                                        ;
0193 3AA330   WRCOM:   LDA    COMSTS    ;Communication card status
0196 E602              ANI    2         ;Check bit 2
0198 CA9301            JZ     WRCOM
0198 7B                MOV    A,E       ;Get character to send
019C 32AFE0            STA    COMDAT    ;Store here
019F C9                RET
                                        ;
01A0 3AAEE0   RDCOM:   LDA    COMSTS    ;Communication card status
01A3 1F                RAR              ;STS bit to carry
01A4 DAB201            JC     READIT
01A7 3A00E0            LDA    APPKBD    ;See if CONTROL-C was typed
01AA FE83              CPI    083H      ;
01AC CA7F01            JZ     DONE
                                        ;
01AF C3A001            JMP    RDCOM     ;No, wait for character
                                        ;
01B2 3AAFE0   READIT:  LDA    COMDAT    ;Get incoming character
01B5 C9                RET
                                        ;
01B6 0E09     PRMSG:   MVI    C,9       ;Print message
01B8 C30500            JMP    BDOS
                                        ;
01BB 0E02     CONOUT:  MVI    C,2       ;Console output
01BD C30500            JMP    BDOS
                                        ;
01C0 436F6D6D61CMDMSG: DB     'COMMAND ERROR$'
01CE 4E6F206469NDSMSG: DB     'NO DIRECTORY SPACE$'
01E1 0D0A446F77DONMSG: DB     '13,10,'DOWNLOAD COMPLETE$'
01F5 446F776E6CWRKMSG: DB     'DOWNLOADING',0
                                        ;
0201                   END
```

Appendix C
SoftCard Version Differences

SoftCard Enhancements

Because of the Apple I/O interface and dual microprocessor environment, the SoftCard implementation of CP/M has the following enhancements:

Patch areas in the BIOS for adding user-written I/O driver software

A screen function interface for modifying the screen attributes for a specific terminal or program

A character redefinition table for redefining the ASCII characters produced by the keys

A type-ahead buffer for keyboard entry while CP/M is performing other operations

A print buffer that allows the printing of a file while performing other operations

80-column display operation is standard

Up to 59.5K bytes of memory for application programs

CP/M Implementation Differences

The SoftCard version does not include the MOVCPM or the SYSGEN utilities. Because the SoftCard implementation of CP/M is a "fixed" size, there is no need for either utility.

SoftCard Differences

The Premium SoftCard IIe System has several features that the previous SoftCards do not have. There are also differences in the way the Premium SoftCard IIe System performs I/O functions.

The following features are unique to the Premium SoftCard IIe System.

64K bytes of RAM which can be accessed by either the Z80 or the 6502 microprocessors

Circuitry for a full 80 columns of display

A type-ahead buffer for keyboard entry while CP/M performs other operations

A print buffer that allows printing of a file while CP/M performs other operations

Differences in Hardware

The SoftCard IIe circuit board contains a Z80B microprocessor which operates three times as fast as those for previous SoftCard circuit boards. The Z80B is not synchronized or phase-locked to the Apple IIe internal clocks.

The Premium SoftCard IIe circuit board is physically larger than the previous SoftCards. It is also designed to install into the AUXILIARY connector slot only.

There are no switches on the Premium SoftCard IIe circuit board.

Z80 interrupts to the 6502 are not allowed; there is no circuit path provided by the Apple IIe.

Differences in Software

The Premium SoftCard IIe has a larger TPA (59.5K bytes) for running programs.

Because all memory is contained on the Premium SoftCard IIe circuit board, there is no need to change the size of CP/M. Therefore, the SoftCard IIe package does not include the CPM60 utility program.

There is no I/O Configuration Block (IOCB). Some of the SoftCard IIe I/O configuration tables and routines are located in different areas of the BIOS and not in a contiguous block.

The screen menus in the CONFIGIO utility program have been changed.

The DOWNLOAD program is provided on the SoftCard IIe Supplement disk instead of Premium SoftCard IIe Master disk. The Microsoft BASIC Interpreter has been condensed into one file (GBASIC.COM). High-resolution graphic commands are available whenever BASIC is running.

Because of the Z80B microprocessor, programs running under the SoftCard IIe version of CP/M execute three times faster than programs running under previous SoftCards.

Differences in I/O Operation

The SoftCard IIe uses a method of accessing the I/O system that differs from the previous SoftCards. The SoftCard Z80 microprocessor uses the 6502 as an I/O processor and the Apple memory for I/O communications. Therefore, it is not possible with the SoftCard IIe version of CP/M to directly access Apple I/O memory locations.

Note

The previous SoftCards access I/O functions directly through memory-mapped locations in the Apple's memory and do not use the 6502 except to call 6502 subroutines.

The SoftCard IIe calls 6502 routines differently than previous SoftCards. The Z80 performs I/O operations through the 6502 microprocessor by accessing a program called the "6502 Basic Input/Output System," or 6502 BIOS. There are 15 separate functions. All are accessed by storing information in a seven-byte area located at 45—4B, and then performing a Z80 CALL instruction to memory location 40. Information from the I/O system is returned to the same seven-byte area.

Index

MICROSOFT™
10700 Northup Way, Bellevue, WA 98004

Software Problem Report

Name _____

Street _____

City _____ State _____ Zip _____

Phone _____ Date _____

Instructions

Use this form to report software bugs, documentation errors, or suggested enhancements. Mail the form to Microsoft.

Category

_____ Software Problem

_____ Software Enhancement

_____ Documentation Problem (Document #_____)

_____ Other

Software Description

Microsoft Product _____

Rev. _____ Registration # _____

Operating System _____

Rev. _____ Supplier _____

Other Software Used _____

Rev. _____ Supplier _____

Hardware Description

Manufacturer _____ CPU _____ Memory _____ KB

Disk Size _____" Density: Sides:

Single _____ Single _____

Double _____ Double _____

Peripherals _____

Problem Description

Describe the problem. (Also describe how to reproduce it, and your diagnosis and suggested correction.) Attach a listing if available.