MICROSOFT® PREMIUM SOFTCARD® IIe

Package

For Apple® IIe

MICROSOFT LICENSE AGREEMENT

CAREFULLY READ ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT PRIOR TO BREAKING THE DISKETTE SEAL. BREAKING THE DISKETTE SEAL INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS.

If you do not agree to these terms and conditions, return the unopened diskette package and the other components of this product to the place of purchase and your money will be refunded. No refunds will be given for products which have opened diskette packages or missing components.

1. LICENSE: You have the non-exclusive right to use the enclosed program. This program can only be used on a single computer. You may physically transfer the program from one computer to another provided that the program is used on only one computer at a time. You may not electronically transfer the program from one computer to another over a network. You may not distribute copies of the program or documentation to others. You may not modify or translate the program or related documentation without the prior written consent of Microsoft.

YOU MAY NOT USE, COPY, MODIFY, OR TRANSFER THE PROGRAM OR DOCUMENTATION, OR ANY COPY EXCEPT AS EXPRESSLY PROVIDED IN THIS AGREEMENT.

2. BACK-UP AND TRANSFER: You may make one (1) copy of the program solely for back-up purposes. You must reproduce and include the copyright notice on the back-up copy. You may transfer and license the product to another party if the other party agrees to the terms and conditions of this Agreement and completes and returns a Registration Card to Microsoft. If you transfer the program you must at the same time transfer the documentation and back-up copy or transfer the documentation and destroy the back-up copy.

3. COPYRIGHT: The program and its related documentation are copyrighted. You may not copy the program or its documentation except as for back-up purposes and to load the program into the computer as part of executing the program. All other copies of the program and its documentation are in violation of this Agreement.

4. TERM: This license is effective until terminated. You may terminate it by destroying the program and documentation and all copies thereof. This license will also terminate if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy all copies of the program and documentation.

5. HARDWARE COMPONENTS: Microsoft product hardware components only include circuit cards and the mechanical mouse.

6. LIMITED WARRANTY: THE PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE PROGRAM IS ASSUMED BY YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU (AND NOT MICROSOFT OR ITS DEALERS) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. FURTHER, MICROSOFT DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OF, OR THE RESULTS OF THE USE OF, THE PROGRAM IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE; AND YOU RELY ON THE PROGRAM AND RESULTS SOLELY AT YOUR OWN RISK.

Microsoft does warrant to the original licensee that the diskette(s) on which the program is recorded be free from defects in materials and workmanship under normal use and service for a period of ninety (90) days from the date of delivery as evidenced by a copy of your receipt. Microsoft warrants to the original licensee that the hardware components included in this package are free from defects in materials and workmanship for a period of one year from the date of delivery to you as evidenced by a copy of your receipt. Microsoft's entire liability and your exclusive remedy shall be replacement of the diskette or hardware component not meeting Microsoft's limited warranty and which is returned to Microsoft with a copy of your receipt. If failure of the diskette or hardware component has resulted from accident, abuse or misapplication of the product, then Microsoft shall have no responsibility to replace the diskette or hardware component under this Limited Warranty. In the event of replacement of the hardware component the replacement will be warranted for the remainder of the original one (1) year period or 30 days, whichever is longer.

THE ABOVE IS THE ONLY WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE THAT IS MADE BY MICROSOFT ON THIS MICROSOFT PRODUCT. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

NEITHER MICROSOFT NOR ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION, PRODUCTION, OR DELIVERY OF THIS PROGRAM SHALL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, OR INCIDENTAL DAMAGES ARISING OUT OF THE USE, THE RESULTS OF USE, OR INABILITY TO USE SUCH PRODUCT EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR CLAIM. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

7. UPDATE POLICY: In order to be able to obtain updates of the program, the licensee and persons to whom the program is transferred in accordance with this Agreement must complete and return the attached Registration Card to Microsoft. IF THIS REGISTRATION CARD HAS NOT BEEN RECEIVED BY MICROSOFT, MICROSOFT IS UNDER NO OBLIGATION TO MAKE AVAILABLE TO YOU ANY UPDATES EVEN THOUGH YOU HAVE MADE PAYMENT OF THE APPLICABLE UPDATE FEE.

8. MISC.: This license agreement shall be governed by the laws of the State of Washington and shall inure to the benefit of Microsoft Corporation, its successors, administrators, heirs and assigns.

9. ACKNOWLEDGEMENT: YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT, AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU ALSO AGREE THAT THIS AGREEMENT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF AGREEMENT BETWEEN THE PARTIES AND SUPERCEDES ALL PROPOSALS OR PRIOR AGREEMENTS, VERBAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN THE PARTIES RELATING TO THE SUBJECT MATTER OF THIS AGREEMENT.

Should you have any questions concerning this Agreement, please contact in writing Microsoft, Customer Sales and Service, 10700 Northup Way, Bellevue, WA 98004.

Microsoft® Premium SoftCard® IIe

System

for Apple® IIe Computer

Installation and Operation Manual

Microsoft Corporation

The Premium SoftCard IIe circuit board was designed for Microsoft by Burtronix of Orange, California.

If you have comments about this documentation or the enclosed software, complete the Software Problem Report at the back of this manual and return it to Microsoft.

Part No. 047-023-003

Document No. 8815-226-00

Preface

Your Microsoft® Premium SoftCard® IIe System is a valuable addition to your Apple® IIe computer. It is your key to running the many programs and languages available under the CP/M® operating system. The Premium SoftCard IIe System is easy to install and use. In addition to the standard CP/M utility programs, it includes several utility programs written by Microsoft and the Microsoft BASIC Interpreter, so you can create your own programs.

Important

Before you break the seal on the disk envelope, you must read the Microsoft License Agreement in the Customer Service Plan booklet. Also before you break the seal on the disk envelope, read the Digital Research License Agreement included with your Customer Service Plan booklet. If you agree with the terms of the agreement, fill out the Microsoft Product Registration Card and mail it to Microsoft immediately.

About the Microsoft SoftCard Manuals

The SoftCard system is documented in four manuals: the *Microsoft Premium SoftCard IIe System Installation and Operation Manual*, the *Microsoft BASIC Interpreter Reference Manual*, the *Osborne® CP/M User Guide*, and the *Microsoft Premium SoftCard IIe System Programmer's Manual.*

The *Microsoft Premium SoftCard IIe System Installation and Operation Manual* introduces the Premium SoftCard IIe package. It also describes how to install the SoftCard IIe circuit board, how to load and use the CP/M operating system, and how to use CP/M built-in commands and certain transient programs. It should be read before installing the SoftCard IIe circuit board and the software.

For programmers who want to connect nonstandard I/O devices or use software requiring modifications to CP/M, the *Premium SoftCard IIe System Programmer's Manual* contains the necessary information. It is also a reference manual for utility programs, commands, and CP/M system calls. This manual can be obtained from Microsoft by sending in your Microsoft Product Registration card.

The *Microsoft BASIC Interpreter Reference Manual* explains how to use Microsoft BASIC and provides a reference for all of the commands, statements, and functions contained in Microsoft BASIC. This manual is intended for users who already know how to program in BASIC. If you are new to BASIC, see the list of recommended reading for more information about programming in BASIC.

If you are new to the CP/M operating system, the *Osborne CP/M User Guide* will teach you how to use the CP/M built-in commands and transient programs. It will guide you step-by-step through the different functions of CP/M.

How to Use This Manual

This is your Premium SoftCard IIe System owner's manual; it shows you how to install and operate your Premium SoftCard IIe System. It is organized so you can find the information you want quickly and easily.

The chapters and appendices are organized as follows:

1 Introduction

Introduces the Premium SoftCard IIe System and lists the syntax notation used in SoftCard IIe documentation.

2 Installation

Describes what is needed to install the Premium SoftCard IIe circuit board and how to do it. This chapter also tells you what other accessory boards are compatible with the SoftCard IIe.

3 Getting Started

Tells you how to load CP/M and lists the procedures for making backup copies of your Premium SoftCard IIe Master disk.

4 An Introduction to CP/M

Introduces the CP/M operating system and describes the role of an operating system within the computer.

5 Using CP/M With the Apple IIe Computer

Describes how CP/M works with the Apple IIe computer. Includes descriptions of the special features of the Premium SoftCard IIe System.

6 CP/M Commands and Utility Programs

Explains how to use CP/M commands and describes the transient programs you will use most often.

7 I/O Configuration

Explains how to configure CP/M for the special requirements of certain I/O devices and application programs.

Appendices

A CP/M Error Messages

Lists and explains the error messages that may be encountered in using CP/M and the utility programs.

B Glossary

Defines terms used in CP/M and SoftCard IIe documentation.

Digital Research License Information

Our license with Digital Research for the CP/M operating system requires that each purchaser of the SoftCard with CP/M register with Microsoft Corporation so that records can be maintained of all CP/M owners. This requirement is made by Digital Research, not by Microsoft. A post card is enclosed for reply. The serial number on the card is the number stamped on the disk labels. Before signing the card and returning it to Microsoft, read the software license agreement below carefully.

Software License Agreement

Important: All Digital Research programs are sold only on the condition that the purchaser agrees to the following license. READ THIS LICENSE CAREFULLY. If you do not agree to the terms contained in this license, return the packaged disk UNOPENED to your distributor and your purchase price will be refunded. If you agree to the terms contained in this license, fill out the REGISTRATION information and RETURN by mail to MICROSOFT CORPORATION.

DIGITAL RESEARCH agrees to grant and the Customer agrees to accept on the following terms and conditions non-transferable and nonexclusive licenses to use the software program(s) (Licensed Programs) herein delivered with this agreement.

Term

This agreement is effective from the date of receipt of the above-referenced program(s) and shall remain in force until terminated by the Customer upon one month's prior written notice, or by Digital Research, as provided below.

Any license under this Agreement may be discontinued by the Customer at any time upon one month's prior written notice. Digital Research may discontinue any license or terminate this Agreement if the Customer fails to comply with any of the terms and conditions of this Agreement.

License

Each program license granted under this Agreement authorizes the Customer to use the Licensed Program in any machine readable form on any single computer system (referred to as System). A separate license is required for each System on which the Licensed Program will be used.

This Agreement and any of the licenses, programs, or materials to which it applies may not be assigned, sublicensed or otherwise transferred by the Customer without prior written consent from Digital Research. No right to print or copy, in whole or in part, the Licensed Programs is granted except as hereinafter expressly provided.

Permission to Copy or Modify Licensed Programs

The Customer shall not copy, in whole or part, any Licensed Programs which are provided by Digital Research in printed form under this Agreement. Additional copies of printed materials may be acquired from Digital Research.

Any Licensed Programs which are provided by Digital Research in machine readable form may be copied, in whole or in part, in printed or machine readable form in sufficient number for use by the Customer with the designated System, to understand the contents of such machine readable material, to modify the Licensed Program as provided below, for back-up purposes, provided, however, that no more than five (5) printed copies will be in existence under any license at any one time without prior written consent from Digital Research. The Customer agrees to maintain appropriate records of the number and location of all such copies of Licensed Programs. The original, and any copies of the Licensed Programs, in whole or in part, which are made by the Customer shall be the property of Digital Research.

This does not imply, of course, that Digital Research owns the media on which the Licensed Programs are recorded. The Customer may modify any machine readable form of the Licensed Programs for his own use and merge it into other program material to form an updated work, provided that, upon discontinuance of the license for such Licensed Program, the Licensed Program supplied by Digital Research will be completely removed from the updated work. Any portion of the Licensed Program included in an updated work shall be used only if on the designated System and shall remain subject to other terms of this Agreement.

The Customer agrees to reproduce and include the copyright notice of Digital Research on all copies, in whole or in part, in any form, including partial copies of modifications, of Licensed Programs made hereunder.

Protections and Security

The Customer agrees not to provide or otherwise make available any Licensed Program including but not limited to program listings, object code, and source code, in any form, to any person other than Customer or Digital Research employees, without prior written consent from Digital Research, except with the Customer's permission for purposes specifically related to the Customer's use of the Licensed Program.

Discontinuance

Within one month after the date of discontinuance of any license under this Agreement, the Customer will furnish Digital Research a certificate certifying that through his best effort, and to the best of his knowledge, the original and all copies, in whole or in part, in any form, including partial copies in modifications, of the Licensed Program received from Digital Research or made in connection with such have been destroyed, except upon written authorization from Digital Research, the Customer may retain a copy for archive purposes.

Disclaimer of Warranty

Digital Research makes no warranties with respect to the Licensed Programs. The sole obligation of Digital Research shall be to make available all published modifications or updates made by Digital Research to Licensed Programs which are published within one (1) year from date of purchase, provided Customer has returned the Registration Card delivered with the Licensed Program.

Limitation of Liability

THE FOREGOING WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL DIGITAL RESEARCH BE LIABLE FOR CONSEQUENTIAL DAMAGES EVEN IF DIGITAL RESEARCH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

General

If any of the provisions, or portions thereof, of this Agreement are invalid under any applicable statute or rule of law, they are to that extent to be deemed omitted.

Contents

Chapter 1
Introduction

The Microsoft Premium SoftCard IIe System is a hardware and software package that greatly enhances the capabilities of your Apple IIe computer. The hardware gives your computer the capability to run CP/M programs, an 80-column display, and an additional 64K bytes of memory. The software in your Premium SoftCard IIe package includes the CP/M operating system, the Microsoft BASIC Interpreter, and special SoftCard IIe utility programs.

Hardware

The circuit board that you receive in your Premium SoftCard IIe package is actually three circuit boards in one. It combines the functions of:

A Z80® coprocessor board

An expansion memory board

An 80-column video display board

By combining the functions of three circuit boards on one board, the Premium SoftCard IIe circuit board saves the remaining accessory slots in the Apple IIe computer for other purposes.

The coprocessor section of the SoftCard IIe circuit board contains a Z80 microprocessor with the interface circuitry necessary for communicating with the Apple IIe I/O bus. A coprocessor is an additional microprocessor which shares control of the computer. Thus, when you install the Premium SoftCard IIe System into your Apple, you are really creating two computers from one: a 6502 computer that will run Apple DOS programs, and a Z80 computer that will run CP/M programs.

The memory section of the SoftCard IIe circuit board contains 64K bytes of RAM (Random Access Memory) for use by either the Z80 or 6502 microprocessors. The memory section permits large application programs (up to 59K bytes) to run under CP/M. The 6502 microprocessor can access the full amount of SoftCard memory to run Apple DOS or Apple Pascal programs.

The 80-column video display section permits an 80-column-wide screen display for CP/M programs. When Apple DOS or Apple Pascal programs are running, the SoftCard circuit board can display either 80 columns or 40 columns.

Once the SoftCard IIe circuit board has been installed, you can operate your Apple IIe computer in either 6502 mode (using the 6502 microprocessor) or CP/M mode (using the Z80 microprocessor). When you are in 6502 mode, the SoftCard will not affect the operation of your Apple IIe. It does allow the 6502 to access the memory and display circuits of the SoftCard without any hardware or software modifications. In CP/M mode, you can run any CP/M-based program or language, including the Microsoft BASIC Interpreter. The 80-column display is standard in CP/M mode.

Software

The Premium SoftCard IIe package includes the CP/M operating system (CP/M-80); the Microsoft BASIC Interpreter; and special transient programs to perform utility functions, such as copying disks and modifying CP/M to your particular system environment.

CP/M Operating System and Programs

The CP/M operating system is one of the most widely implemented 8-bit operating systems in use today. Because of its widespread use, an extensive library of high-level languages and application software is available for CP/M-based computers.

In addition to supporting a wide variety of software, CP/M offers many convenient features. These include the capability of implementing machine-language programs; faster disk I/O access; simple file transfer operations; and "wild card" file naming conventions that allow you to refer to multiple files with one name. The SoftCard version of CP/M also includes several programming tools for program development and utility programs for everyday operation.

Microsoft BASIC Interpreter

The Microsoft BASIC Interpreter is the most widely implemented BASIC in use today. In addition to the standard BASIC commands and statements, the SoftCard version of BASIC includes high-resolution graphics commands and statements. The *Microsoft BASIC Interpreter Reference Manual* describes how to use Microsoft BASIC.

Premium SoftCard IIe Utility Programs

Utility programs perform certain time-consuming tasks, such as disk formatting and file transfer. In addition to the CP/M utility programs PIP and STAT, the Premium SoftCard IIe System includes programs that allow you to:

Copy Apple DOS files to your CP/M disk with the APDOS program.

Create startup disks with the AUTORUN program.

Display an alphabetical list of disk files and other information with the CAT program.

Configure CP/M for specific I/O devices and programs with the CONFIGIO program.

Format and copy disks with the COPY program.

Copy files in a single-drive system with the MFT program.

Several other utility programs and programming tools are also included. These are listed in Chapter 6, "CP/M Commands and Utility Programs."

Command Line Notation

Before you can use the computer, you must learn how to communicate with it. Communication is more than just typing words on the keyboard. Instructions to the computer must be in a certain format to ensure the computer understands exactly what you want it to do.

Most instructions to the computer are in the form of *commands*. Commands often consist of a keyword and command line. The keyword is usually the name of the command that you want to execute, such as COPY. The command line that follows the keyword specifies what the command must do. For example,

```
COPY B:=A:
```

instructs the COPY program to copy the contents of the disk in drive A: to the disk in drive B:. Without the command line, COPY would not know what to copy.

In the Premium SoftCard IIe documentation, special notation has been developed to show the differences between what you enter on the keyboard and what you see in the manual. The following notation is used in this manual to help you understand how commands are entered into the computer.

ital	Italics indicate information that you enter. Italicized lowercase text is for an entry that you must supply, such as a *filename*.
[ ]	Square brackets indicate that the enclosed entry is optional. In the "Examples," the /F and /V entries can be included at your discretion. They are not necessary to execute the command.

{ } Braces indicate a choice between two or more entries. At least one of the entries enclosed in braces must be chosen, unless the entries are also enclosed in square brackets. In the COPY example, you must type either the /S or the /D entry in the command line.

| Vertical bars separate choices within braces.

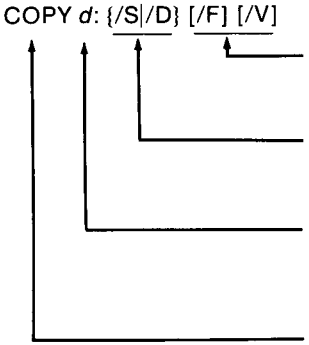
. . . Ellipses indicate that an entry can be repeated as many times as needed or desired. In the following MFT example, the ellipses indicate that you can include additional files in the command line.

CAPS Capital letters not enclosed in the other elements of syntax indicate portions of commands that must be entered exactly as shown, such as command keywords. Small capital letters indicate that you must press a key named by the text; for example, "press the RETURN key."

All other punctuation, such as commas, colons, slash marks, and equal signs, must be entered exactly as shown. In the COPY example, the / character must be included with the entry. Unless specified in the description of the command, spaces are optional. Spaces are usually shown in the documentation for clarity.

Examples

Command Line

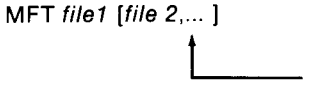


These two entries are optional. They should be typed as shown.

You must type either the /S or the /D entry.

The lowercase italic *d*: means you must supply the disk drive identifier (A: through D:).

Capital letters indicate that the word must be entered exactly as shown.



Ellipses indicate that you can enter additional arguments.

Chapter 2
Installation

This chapter gives instructions for installing the Premium SoftCard IIe circuit board. It also provides other information about setting up your Premium SoftCard IIe System. We recommend reading this entire chapter before installing the SoftCard IIe circuit board.

Preliminary Information

Before installing the circuit board, make sure that your system meets all the criteria listed in the following "System Requirements" section and that your Apple computer is set up and operational as described in the *Apple IIe Owner's Manual.*

System Requirements

To use your Premium SoftCard IIe System successfully, you will need the following items:

1. An Apple IIe computer.
2. A Premium SoftCard IIe printed circuit board.
3. Disk drives—only one drive is needed, but two drives are recommended.
4. A screen monitor or an external terminal.
5. The Premium SoftCard IIe CP/M Master disk.
6. At least three blank floppy disks.

Note

If you want to use an external terminal (a terminal with its own keyboard and display) instead of the Apple IIe keyboard and screen monitor, be sure to read the section in Chapter 7 entitled "Screen Function Interface."

Unpacking

Upon receipt of your Premium SoftCard IIe package, check carefully for missing items or shipping damage. If any of the following items are damaged or missing, contact your computer dealer.

Your Premium SoftCard IIe System should consist of the following:

The Premium SoftCard IIe printed circuit board

The Premium SoftCard IIe Master floppy disk, which includes the CP/M operating system and the following files:

APDOS.COM	DDT.COM	MFT.COM
ASM.COM	DUMP.COM	PATCH.COM
AUTORUN.COM	DUMP.ASM	PIP.COM
BOOT.COM	ED.COM	STAT.COM
CAT.COM	GBASIC.COM	SUBMIT.COM
CONFIGIO.BAS	LOAD.COM	XSUB.COM
COPY.COM		

The *Osborne CP/M User Guide*

A green Microsoft binder containing the following manuals:

Microsoft Premium SoftCard IIe System Installation and Operation Manual

Microsoft BASIC Interpreter Reference Manual

A Customer Service Plan (in the back of the binder)

Disk drive labels

If reshipment of the Premium SoftCard IIe System is necessary, contact Microsoft Corporation prior to returning the circuit board.

Safety and Handling Precautions

The following paragraphs contain some common-sense precautions you should be aware of before attempting installation.

Safety Precautions

Before installing the Premium SoftCard IIe (or any other circuit board), you should know about the possible shock hazards that are present in any electronic device, including a personal computer. Although the Apple computer is designed to minimize the danger of electrical shock, it is prudent to exercise caution whenever the cover has been removed from the computer.

It is dangerous to open any electrical or electronic device while the power is on. Attempts to insert or remove circuit boards while power is on will usually result in damage to the board and to the computer. Use standard electrical safety precautions whenever the cover is off the computer.

Handling Precautions

The Premium SoftCard IIe circuit board contains integrated circuits (called "ICs" or "chips") which can be damaged by electrostatic discharge during handling. "Handling" is defined as physically holding or moving the circuit board outside the computer. The only time you should be handling the circuit board is when you are installing it.

To decrease the chance of damaging the circuit board (and possibly voiding the warranty), follow these simple guidelines when handling the SoftCard IIe or any other circuit board.

1. Discharge any personal static electricity before handling accessory boards by touching the metal power supply chassis.

   Make sure that the power is turned off and the power cord is disconnected before touching the power supply.

2. Hold the circuit board by its edges to avoid contaminating it with oil from your hands. Be especially careful to avoid touching the gold edge-connector on the bottom of the board.

3. When handling circuit boards or ICs, avoid any contact with plastic, vinyl, and styrofoam. These substances can cause electrostatic buildup.

4. If you need to remove or replace ICs, keep handling to a minimum. Take the ICs from their containers only when necessary. Always use anti-static containers for storage. Also, always handle ICs by the casing instead of by the connector pins. This will minimize the electrostatic shock danger and will prevent possible damage to the pins.

Other Accessory Boards

The CP/M operating system requires accessory boards be installed into specific slots, depending on their intended use. For example, if you have a printer interface board, it should be installed in slot 1. This allows you to refer to the printer without specifying a slot number, as is necessary with Applesoft BASIC and Integer BASIC. The accessory slots are numbered from 1 through 7, as shown in Figure 2.1. To determine the correct slot for using other accessory boards with CP/M, read the following paragraphs.

Note

The accessory slot assignments for CP/M are exactly the same as those for Apple Pascal. Therefore, if you have your system configured for use with Apple Pascal, no rearrangement is necessary.

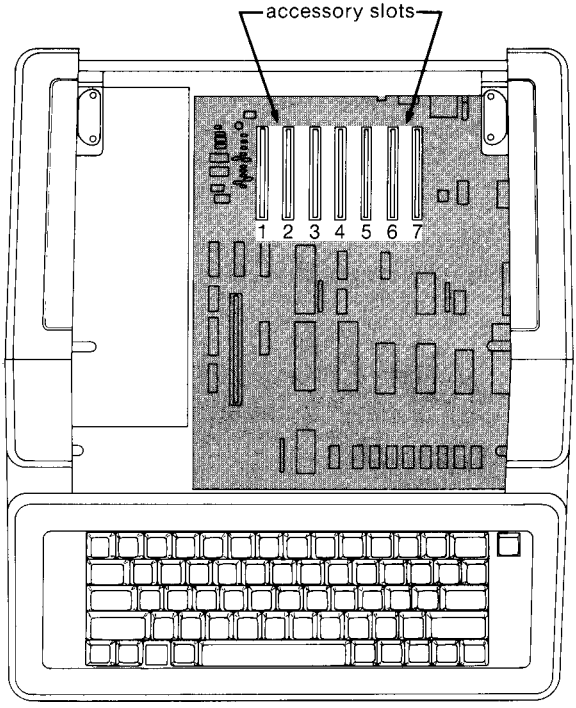


Figure 2.1. Accessory Slots

Compatible Accessory Boards

Table 2.1 lists the accessory boards that are directly compatible with the Premium SoftCard IIe System and CP/M. These boards, when installed in the appropriate Apple accessory slots, will work without any software modifications. (An underlined "x" indicates the recommended slot for installation.)

Table 2.1
Compatible Accessory Boards

	Assigned Slot						
Accessory Board	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Apple Disk II Controller					x	x	
Apple Communications Interface	x	x	x	x			
California Computer Systems® (CCS) 7710A™ Serial Interface*	x	x	x	x			x
Apple High Speed Serial Interface	x	x	x	x			x
Apple Silentype® Printer Interface	x	x	x	x			x
Apple Parallel Printer Card	x			x			x
Apple Firmware Card	x	x	x	x			x
Apple Super Serial Interface	x	x	x	x			x

* The CCS 7710A Serial Interface is the preferred interface board for serial communications. It supports standard communication protocols and variable baud rates from 110 to 19200 baud. The Apple Communications Interface board requires hardware modifications for use with baud rates other than 110 or 300 baud.

There are some accessory boards not listed in Table 2.1 that are compatible with CP/M. As a general rule, any accessory board or I/O device that is directly compatible with the Apple Pascal operating system without requiring any software modifications will be compatible with CP/M as well. Other accessory boards or I/O devices can be used if the software supplied by the board manufacturer can be configured to your CP/M system with the CONFIGIO program. Instructions on how to use CONFIGIO are given in Chapter 7.

Apple Disk Controller Boards

The SoftCard IIe version of CP/M communicates with a maximum of four disk drives. (Two disk drives can be connected to each Apple Disk II Controller board.) As indicated in Table 2.1, Apple Disk II Controller boards can be installed in slots 5 and 6. The first controller board must be installed in slot 6 and the second in slot 5.

Apple disk drives are designated with numbers. The first drive is drive 1, the second drive is drive 2, and so on. CP/M uses letters followed by a colon to identify the disk drive. The first drive (connected to the disk controller in slot 6) is always assigned as drive A:. Each sequential drive is assigned the next successive letter. For example, drive 2 is drive B:, and drive 3 (connected to the disk controller in slot 5) is drive C:.

General Purpose I/O Accessory Boards

Accessory boards for general purpose I/O devices (such as modems, paper-tape readers, and punches) must be installed in slot 2. Any board that is compatible with Apple Pascal or has accompanying interface software (from the board's manufacturer) is compatible with CP/M.

External Terminals

If you are using an external terminal and want to use CP/M, we recommend you use either a CCS 7710A Serial Interface board or a modified Apple Communications Interface board to connect the terminal to your Premium SoftCard IIe System. The Apple High Speed Serial Interface board is compatible but is not recommended, since there is no way for CP/M to check the "status" of this device. The preferred accessory slot for the external interface board is slot 3.

Important

Before using an external terminal with CP/M, certain portions of CP/M must be modified with the CONFIGIO program. Do not install any circuit board into slot 3 until the I/O configuration process is completed. Once CP/M is set up for the external terminal, you may then install the interface board. See "Configuring the Screen Function Interface" in Chapter 7 for more information.

Circuit Board Installation Procedure

Use the following procedure to install the Premium SoftCard IIe circuit board. We recommend that you first read all the instructions to acquaint yourself with the overall procedure. Then perform each step with care exactly as described.

1. Set the Apple IIe POWER switch to OFF (see Figure 2.2).

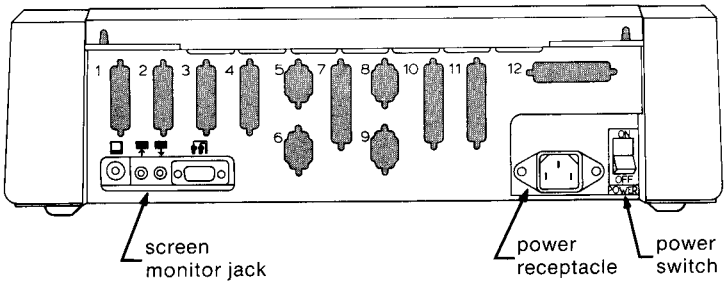


Figure 2.2. Apple IIe Rear Panel

2. Set all connected accessory external power switches to OFF (display monitor, printer, and other external devices).

3. Unplug the power cord from the wall outlet.

4. Ensure that there is nothing on the top cover of the computer.

5. Position your Apple computer with the keyboard directly in front of you, then grasp the tabs at the back of the top cover, as shown in Figure 2.3.

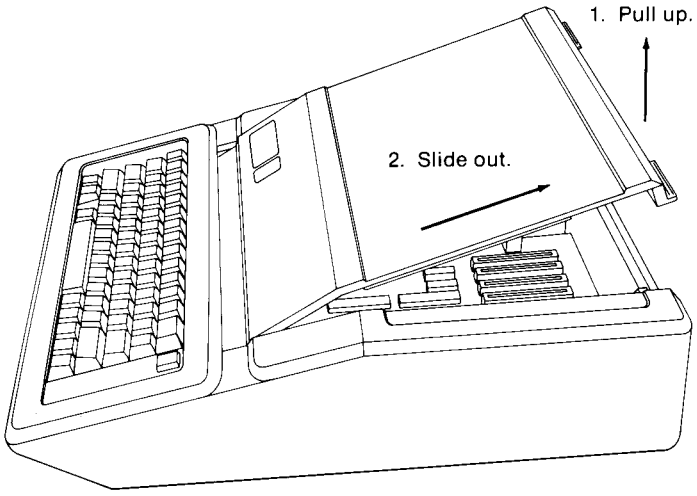


Figure 2.3. Top Cover Removal

6. Pull the cover up gently, until it pops loose.

7. Slide the cover back and remove it from the computer. Once the cover has been removed, place it somewhere out of your way.

8. Using Figure 2.4 as a guide, locate the AUXILIARY CONNECTOR slot. If there is an accessory board in that slot, remove the board and store it in a safe place.

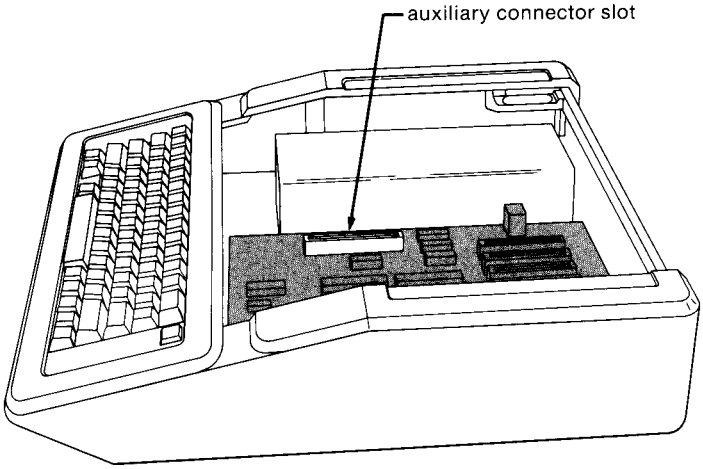


Figure 2.4. Auxiliary Connector Slot

9. Remove the SoftCard IIe circuit board from its container.

10. Align the SoftCard IIe circuit board in the AUXILIARY CONNECTOR slot so that the edge-connector is directly over the connector in the slot. The component side should face the right side of the chassis. Use Figure 2.5 as a guide.

11. Press the circuit board down into the connector using steady but firm pressure. Make certain that the SoftCard circuit board is level and not tilted down toward the front of the Apple.

12. Refer to Table 2.1 for the correct installation configuration of your other accessory boards. See Figure 2.1 for accessory slot locations. If you have any other accessory boards to install, do it now.

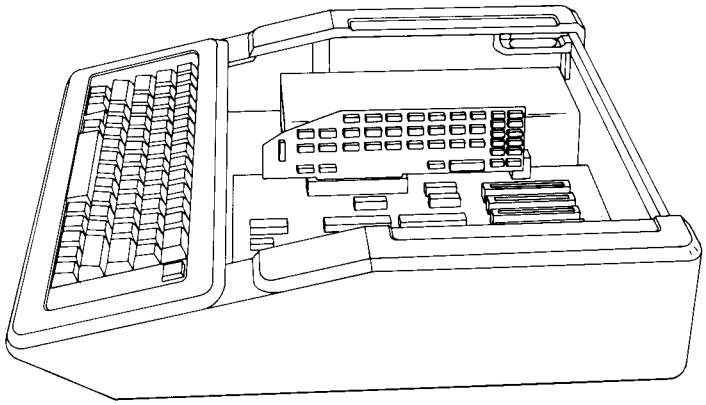


Figure 2.5. Installation of SoftCard IIe

13. Carefully place the top cover back on the computer.

14. Slide the forward edge of the cover under the forward lip of the chassis, as shown in Figure 2.6.

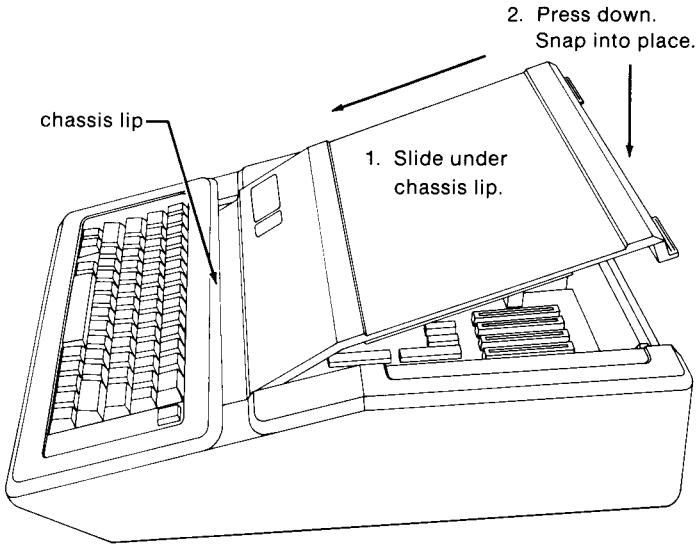


Figure 2.6. Top Cover Replacement

15. Press the rear portion of the cover down until the corners pop back into place.

16. Plug the power cord back into the computer and then into the wall outlet or other power source (a power strip, for example).

17. Turn all other external device power switches (printer, monitor, etc.) to ON.

This completes the circuit board installation procedure. Since CP/M uses letters to identify disk drives, we recommend applying the labels from the SoftCard IIe carton to the front of each disk drive, as indicated:

Drive 1 is labeled drive A:

Drive 2 is labeled drive B:

and so on. Each successive drive is labeled with the next letter.

The hardware of your Premium SoftCard IIe System is now ready to use. The next step is to "install" the software and start using your SoftCard IIe System. The next chapter will show you how.

Chapter 3
Getting Started

Before using a CP/M program, three steps must be taken. These are:

Loading the CP/M operating system into memory

Making backup copies of your SoftCard Master disk

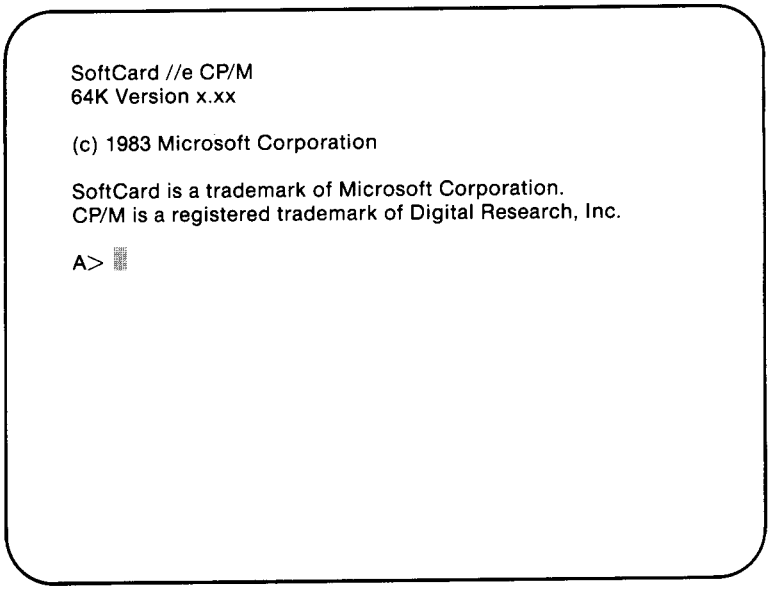
Making any adjustments to CP/M required by your program or an I/O device

This chapter will tell you how to do the first two steps and refers you to the appropriate section for the last step. If you are new to CP/M, you should read this chapter completely before using CP/M.

Loading CP/M

CP/M can be loaded into memory using one of two methods. The first method is to simply insert the Premium SoftCard IIe Master disk into disk drive A: and set the POWER switch to ON. (CP/M disks are inserted into the Apple II disk drives the same way Apple DOS disks are inserted. That is, the disk label faces up and the write-protect notch is on the left.) When you turn on the power, CP/M is automatically loaded into memory. (This process is known as "booting.")

After a few seconds, the screen displays:



The second method is used if the power is already turned on. If you have an Apple disk in disk drive A:, remove it and insert the SoftCard Master disk. Then, while holding down the OPEN-APPLE and CONTROL keys, press the RESET key. As in the first method, this will load and run CP/M. The screen display will be the same.

The A> on your display is a prompt indicating that CP/M is ready to accept a command.

As noted in the previous chapter, CP/M uses letters to identify disk drives. The A> indicates that all disk operations will be performed from drive A:, unless you command otherwise. (See the "d:" command in Chapter 6.)

To see if CP/M is operational, type

```
CAT
```

and press the RETURN key. The CAT command is similar to the Apple DOS CATALOG command, but returns slightly different results. You will see the following display:

```
APDOS    .COM  2K  | CONFIGIO .BAS  7K  | ED      .COM  7K  | PIP     .COM  8K |
ASM      .COM  8K  | COPY     .COM  2K  | GBASIC  .COM  26K | STAT    .COM  6K |
AUTORUN  .COM  1K  | DDT      .COM  5K  | LOAD    .COM  2K  | SUBMIT  .COM  2K |
BOOT     .COM  1K  | DUMP     .COM  5K  | MFT     .COM  2K  | XSUB    .COM  1K |
CAT      .COM  1K  | DUMP     .ASM  1K  | PATCH   .COM  1K  |

Total of 113K bytes in 20 files, 13K bytes available
```

When you typed *CAT* and pressed RETURN, the two actions instructed the CP/M operating system (which is now in the Apple IIe's memory) to find the program CAT on the disk drive and execute the instructions contained within. By doing this, you effectively tested the different software modules of CP/M and the transient program CAT. CAT also shows you the files you have on disk, how much memory space you have left on your Premium SoftCard IIe Master disk, and the size of each of the files. Check what you see on the screen with the display above; they should match.

When you have verified that your SoftCard Master disk has the same files as shown in the display, the next step is to make backup copies of your SoftCard Master disk.

Backing Up the SoftCard Master Disk

A "backup copy" is a duplicate copy of a program or set of programs, usually on a floppy disk or magnetic tape. Making backup copies is a good programming practice that ensures that you will always have a copy of your software in case the original copy is damaged or erased, or just wears out.

Making a backup copy also permits you to modify CP/M for different software and I/O configurations. You should always work with the backup copy and never with the SoftCard Master disk.

Making backup copies of your Premium SoftCard IIe Master disk is a simple, one-step process with the SoftCard COPY program. (The COPY program is fully documented in Chapter 6 of this manual.) When you have made backup copies, store the SoftCard Master disk in a safe, dry place away from magnetic interference.

Use the procedure designed for your system configuration (single-drive or multiple-drive) to make backup copies of the SoftCard Master disk. Both procedures assume that CP/M has been loaded into memory and that the A> prompt is on the screen.

Note

If you make a mistake when typing a command, use the ← (left-arrow) key to backspace and correct it. A description of CP/M line editing commands can be found in Chapter 5 of this manual.

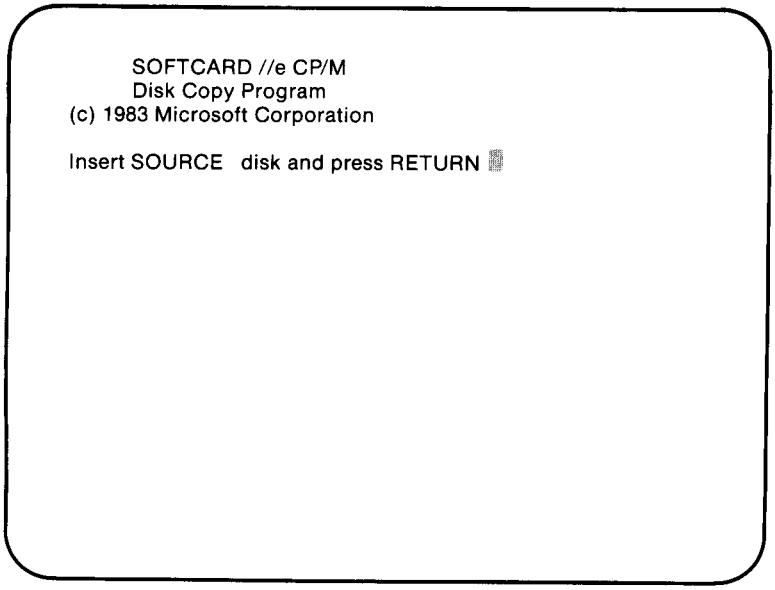
Making Backup Copies With a Single-Drive System

For single-drive systems, use the following procedure to make a backup copy of your Premium SoftCard IIe Master disk. The COPY program will format the backup disk as it copies it.

1. Insert the SoftCard Master disk in drive A:, type

   ```
   COPY A:=A:
   ```

   and press the RETURN key. After a few seconds, the screen displays:



2. Leave the SoftCard Master disk in drive A: and press the RETURN key. The program will then copy a portion of the SoftCard Master disk. The COPY program responds by displaying:

   ```
   Insert DESTINATION disk and press RETURN
   ```

3. Remove the Premium SoftCard IIe Master disk and insert a blank disk into drive A:. Then press the RETURN key.

Note

For single-drive systems, all data copied from the source disk to the destination disk must be held in memory while you change disks. Because the Apple IIe memory is smaller than the amount of data to be copied, only part of the data can be copied at a time.

When the program has written a portion of the disk into memory, it prompts you with:

```
Insert    SOURCE    disk and press RETURN
```

Remove the destination disk (the disk you just copied data to) and insert the SoftCard Master disk back into the disk drive. Press the RETURN key. This process will be repeated several times until the entire disk has been copied. When the copy process is complete, the screen displays:

```
Operation completed

Do you wish to repeat this operation?
```

4. You now have a backup copy of your SoftCard Master disk. To make additional copies, press the *Y* key. If you do not want to make another copy, press the *N* key. The screen displays:

```
Insert CP/M system disk into drive A:

Press RETURN
```

Since you already have a copy of your SoftCard Master disk in the drive, simply press the RETURN key. This will return you to CP/M command level, and you will see the A> prompt.

Making Backup Copies With a Multiple-Drive System

For multiple-drive systems, use the following procedure to make a backup copy of your Premium SoftCard IIe Master disk. The COPY program will format the backup disk as it copies.

1. Insert your SoftCard IIe Master disk into drive A: and type

   ```
   COPY B:=A:
   ```

   and then press the RETURN key. After a few seconds, the screen displays:

```
      SOFTCARD //e CP/M
      Disk Copy Program
(c) 1983 Microsoft Corporation

Insert SOURCE      disk into drive A:
Insert DESTINATION disk into drive B:

Press RETURN to begin
```

2. Insert a blank disk into drive B:. Press the RETURN key to start the copy process.

   When the copy process is complete, the screen will display the message:

   ```
   Operation completed

   Do you wish to repeat this operation?
   ```

3. You now have a backup copy of your SoftCard Master disk. If you want to make another copy, press the *Y* key and follow the instructions given on the screen.

   When you have finished copying disks, press the *N* key. The program responds with:

   ```
   Insert CP/M system disk into drive A:

   Press RETURN
   ```

   Remove the SoftCard Master disk and store it in a safe, dry place away from magnetic interference.

4. Remove the backup copy you have just made from drive B: and insert it into drive A:. Press the RETURN key to exit to CP/M command level.

I/O Configuration

If you are using nonstandard I/O devices, the final step in getting started with CP/M is modifying the I/O portion of CP/M. (See "Compatible Accessory Boards" in Chapter 2 for information on nonstandard devices.) If you are using standard devices, you can start using your Premium SoftCard IIe System now.

The CP/M operating system in your SoftCard IIe package is configured internally to work with most standard Apple II and IIe accessories. However, you may have to modify the I/O portion of CP/M to accommodate some of the accessories that

are not directly compatible with CP/M. This is particularly true if you are using:

An external terminal

An 80-column video display board (other than SoftCard) for different character fonts on the screen

Nonstandard I/O "driver" software

A modem for telecommunications

A different disk drive system, such as a "hard disk"

Most of the modifications to CP/M can be made with the CONFIGIO program. For more information, read the following sections:

To use an external terminal or an 80-column video display board with your system, read the "Screen Function Interface" section in Chapter 7.

If your terminal requires additional software to run with CP/M, read "Nonstandard I/O Devices and User Software" in Chapter 7 also.

To add additional I/O software, read "Nonstandard I/O Devices and User Software" in Chapter 7.

To use a different disk drive system with your Apple IIe (other than the Disk II drives), you will probably need additional software. The manufacturer of the disk drive system will usually provide explicit instructions for modifying CP/M for the disk drive system. Check with your computer dealer before installing non-Apple disk drives into your system.

Chapter 4
An Introduction to CP/M

This chapter explains in general terms how the CP/M operating system works with your Apple IIe computer. Reading this chapter is not necessary for running application programs, but it is recommended for a better understanding of how the components of a computer system work together. If you are new to CP/M, we recommend you read the *Osborne CP/M User Guide* after completing this chapter.

Components of a Computer System

In order to understand how CP/M works, it is necessary to understand how computers work in general. This section describes the different components of a computer system and explains how they work together.

Computers are used for a wide variety of purposes, from scientific and business applications to home entertainment. All applications, regardless of their purpose, perform one common function: processing data (information) for a desired end result. For example, a business program may process the figures that are part of a debit account and return those figures in an accounts receivable format. A game, on the other hand, may take the data supplied by your hand movements (through a game-controller) and process that data to move a figure on the screen. To accomplish either goal, the computer processes the data through the various components that comprise the computer system.

A computer system has many components that are divided into two general categories: hardware and software. Both are necessary to process data.

Hardware

Hardware is the term that is applied to the physical components of the computer system—the keyboard, screen monitor, accessory boards, and I/O devices (printers, disk drives, and so on). The most important hardware components are the CPU, memory, and the input/output interface.

The CPU (central processing unit) is a device called a *microprocessor*. A microprocessor is an integrated circuit (also called an IC or "chip") that performs the actual processing of data ("computing") by executing instructions stored in the computer's memory.

There are many types of microprocessors, and they vary in how much data they can process and how fast they can process it. The Apple IIe computer uses the *6502* microprocessor as its CPU. Other microprocessors include the Zilog Z80 and the Intel® 8080. Each of these microprocessors has its own instruction set, which is simply the total repertoire of commands that the CPU will recognize and execute.

The second major component of a computer is *memory*. Memory is where programs and data are stored. The Apple IIe uses two types of memory: *internal memory* for storing programs and data for immediate execution; and *mass external storage* (usually a disk drive) for storing files and large amounts of data that are not needed by the computer for immediate execution.

Internal memory is either RAM (random access memory) or ROM (read only memory). RAM is used when data or programs must be stored and revised easily. It is usually the largest portion of the internal memory in a typical computer. ROM, on the other hand, is used for short programs which are never revised at all, such as a *bootstrap loader* (also called a *boot program*). A bootstrap loader is an initializing program which loads other programs into memory. It is executed from ROM when power to the computer is turned on.

The third major hardware component of the computer is a set of circuits collectively known as the input/output (I/O) interface. The I/O interface includes the *I/O Bus* (the circuit paths between the CPU and the accessory slots) and any interface printed circuit boards that are installed in the accessory slots. The interface boards connect to external I/O devices, such as, printers, terminals, or disk drives. Although the keyboard on the Apple IIe appears to be an integral part of the computer, it is really an I/O device for operator input.

Software

The *software* components control the actions of the computer. The software components you will use most often are called *programs*. Programs are defined as a set of instructions that tell the computer to perform a certain task in a specified manner. Software generally means a "program" but can also include simple machine instructions. Programs are further divided into routines, subroutines, statements, and instructions.

The fundamental building block of all software components is the *machine instruction*. A machine instruction is a coded number that the CPU recognizes as a command to perform a low-level or "primitive" task, such as sending a character to memory. Because it is difficult to communicate with the CPU in machine instructions, they are organized into blocks of instructions, called assembly language instructions.

Assembly language is a low-level programming language which uses mnemonic symbols to indicate what each CPU instruction does when executed. Note that there is no one language called "assembly language" as there is a language called FORTRAN. Assembly language is a generic term for a microprocessor's instruction set that can be used for programming. Therefore, each microprocessor has its own assembly language. To make writing programs easier, assembly language instructions are organized into larger blocks for use in high-level languages.

High-level languages are programming languages that use English-like statements for instructions. For example, the statement "PRINT" is an instruction to print a character (or characters) on the screen. It corresponds to several assembly language instructions, which in turn correspond to several machine instructions. The term "high-level" refers to the degree of complexity in how the language is structured and the amount of memory required to run it. BASIC, FORTRAN, COBOL, and LISP are just some of the high-level languages available. BASIC is the most commonly used high-level language for microcomputers.

Application programs are programs that perform a certain set of functions associated with a specific task, such as word processing. Application programs can be written either in a high-level language or in assembly language. They form the final level of translation between the user and the CPU. Most application programs use normal English sentence structure and menus.

The Role of an Operating System

An operating system is a "program" that coordinates the different components of a computer system and provides you with a direct way of controlling the computer. You could control your computer system without an operating system by using assembly language instructions, but doing so would be very time-consuming and tedious. An operating system performs three functions:

1. It controls the activities of the disk subsystem (the disk drives, interface circuits, and disk software). An operating system manages the storage and retrieval of data from these files and monitors the location and the amount of memory each file occupies.

2. It provides a convenient means for loading and executing programs from storage devices, such as disk drives.

3. It controls the activities of the I/O subsystem. Through an operating system, you can control the flow of data between the CPU and I/O devices such as the terminal, printer, and disk drives.

For an operating system to perform these functions, it must be able to communicate with the CPU and the other components of the computer system. Since different microprocessors have different instruction sets for communication, an operating system must be written for a specific microprocessor.

CP/M

CP/M (Control Program/Microprocessors) is an operating system written for the 8080 and Z80 microprocessors, but not for the Apple 6502 microprocessor. The reason is the incompatibility of the 6502 instruction set. (There are two operating systems written for the 6502 microprocessor: Apple DOS and Apple II Pascal.)

CP/M was originally written for the Intel 8080 microprocessor. The Z80 microprocessor, which was introduced later, has a very similar instruction set and is compatible with CP/M. Many computers use either the 8080 or the Z80 microprocessor and thus share a common means for programming. This is the primary reason why many computers use CP/M for an operating system.

SoftCard, CP/M, and the Apple IIe

To run CP/M programs on the Apple IIe computer, an 8080 or Z80 microprocessor is needed. The SoftCard circuit board contains a Z80 microprocessor and the interface circuitry for running CP/M programs.

The Z80 microprocessor on the SoftCard allows you to run CP/M programs whenever the CP/M operating system is loaded into memory from disk. The CP/M bootstrap loader has been modified by Microsoft to activate the Z80 microprocessor. Because all CP/M instructions are written for the Z80, all commands from the keyboard are executed by the Z80. The Z80, however, uses the 6502 microprocessor to process I/O data to and from the I/O subsystem.

Note

When you load the Apple DOS or the Apple Pascal operating systems into memory, the 6502 will execute all instructions. The 6502 does not use the Z80 for any purpose. The Z80 therefore remains in a "wait" state until it receives a command that is part of its instruction set.

CP/M Bootstrap Loader

As explained in the previous section, an operating system is the "program" that, among other functions, permits the computer to gain access to data and other programs from a mass storage device, such as a disk drive. Because an operating system cannot be in memory to load itself, however, another program must perform this function. This is done with a program called a bootstrap loader.

The bootstrap loader in the Apple IIe is an assembly language routine in ROM that reads the first sector of the first track of a disk drive into the lowest memory locations every time power is turned off and then on, or when CONTROL-RESET is pressed. When a CP/M disk is in drive A:, the Apple bootstrap loader loads into memory the data that is stored on Track 0, Sector 1 of the disk. ("Tracks" and "sectors" are explained in the "Disk Organization" section of this chapter.) This data contains a second bootstrap loader. When loaded into memory, the second bootstrap loader in turn loads the rest of CP/M. This process is shown in Figures 4.1 and 4.2. Once CP/M has been loaded into memory, it uses other loader routines to load CP/M programs.

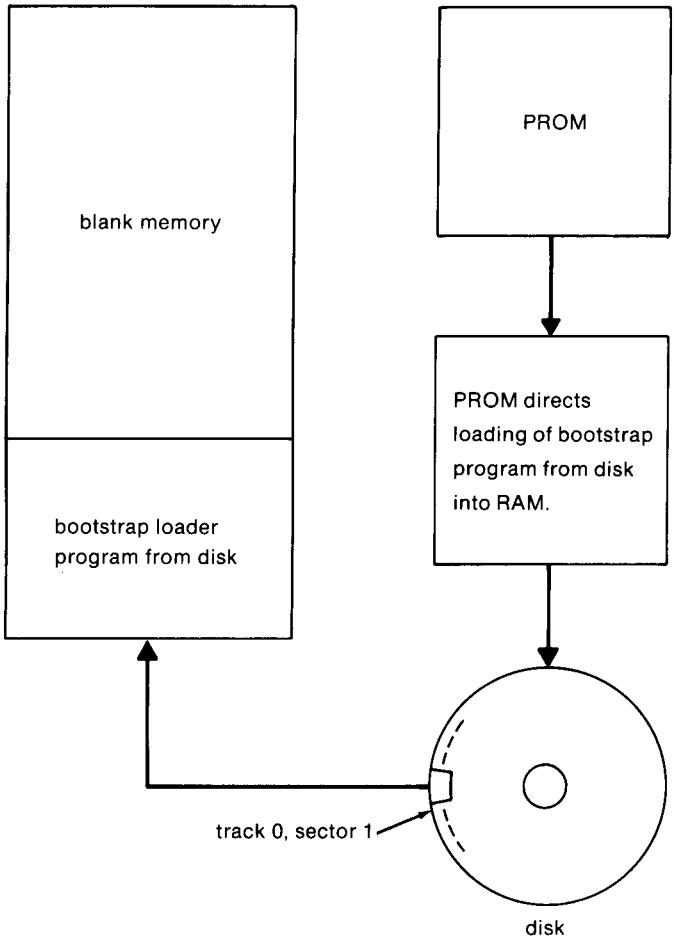


Figure 4.1. Memory Locations

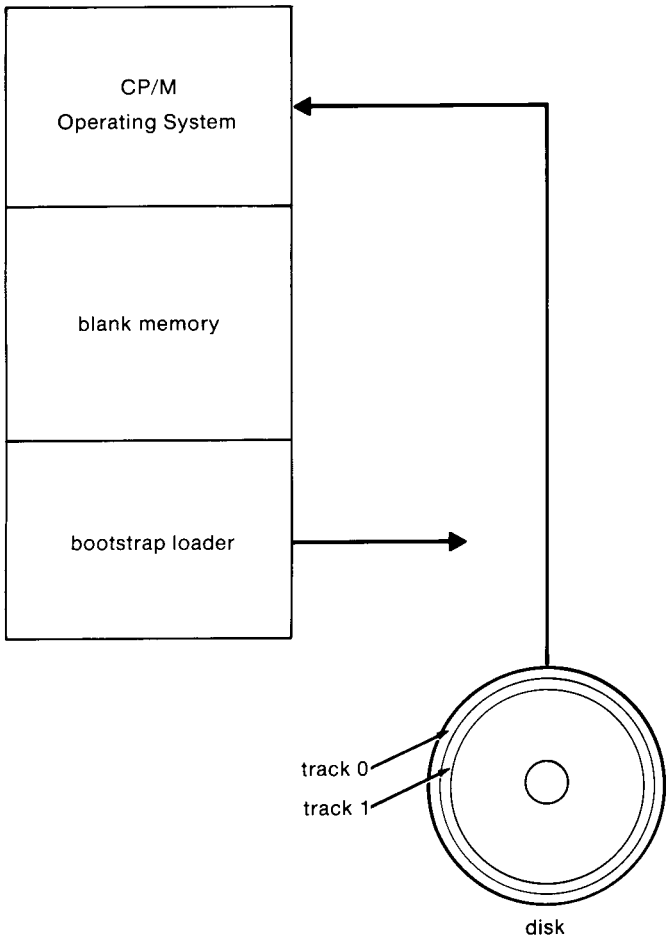


Figure 4.2. The Bootstrap Loader

The process of loading and starting the operating system is called "booting." CP/M can be booted by one of two methods. The first is a *cold start*, which loads the entire operating system into memory after the power has been turned off and then back on. Cold starts are performed whenever you want to reload the entire operating system. For example, if you were using Apple DOS and wanted to use a CP/M program, you would first remove the Apple DOS disk from drive A: and replace it with a CP/M system disk. Then, you would perform a cold start by pressing the RESET key while holding down the CONTROL and OPEN-APPLE keys.

The other booting method is a *warm start*. A warm start is performed by pressing CONTROL-C. The difference between the two is that a cold start reads the whole CP/M operating system into memory while a warm start reads in only a portion of CP/M. The rest of CP/M is assumed to be intact since there has been no loss of power. The programs, if operating properly, will not alter the memory containing the other part of the CP/M operating system. Warm starts are used whenever you change disks in the active drive; or when you need to clear an error condition.

How CP/M Uses Memory

The CP/M operating system that you receive with the Premium SoftCard IIe System consists of two types of software: sets of assembly language routines organized into modules, and executable programs called transient programs. (Transient programs are explained in the "CP/M Transient Programs" section in this chapter.) The software modules that are loaded into memory form the nucleus of CP/M. These modules perform the functions of an operating system. (See "The Role of an Operating System" in this chapter.)

When you load CP/M with a cold start, you are loading only the software modules into memory. The modules are loaded into specific areas of memory, as shown in Figure 4.3. The lowest section of memory is reserved for the bootstrap loader routine first, and then any other software necessary for performing warm and cold starts. (Note that the TPA is not a software module, but a dedicated area of memory for programs.) The three CP/M software modules are the CCP, BIOS, and BDOS.

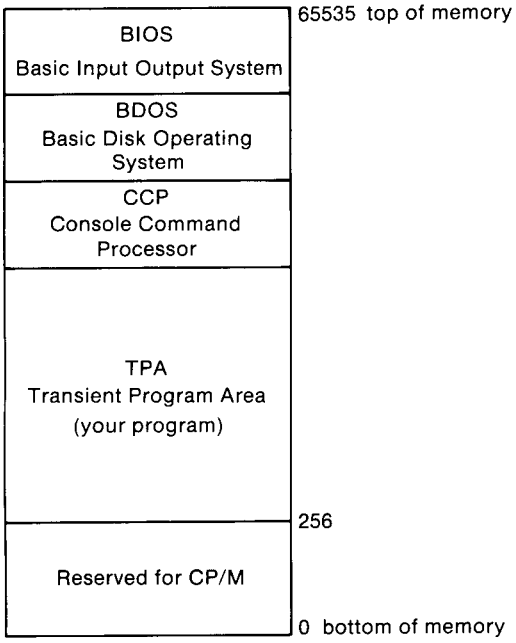


Figure 4.3. Software Module

The *TPA* (Transient Program Area) is where programs or languages are stored and executed under CP/M. For the SoftCard version of CP/M, programs up to 59K bytes in size can run in the TPA.

The *CCP* (Console Command Processor) is the software module which controls the interaction between you and the computer at CP/M command level. The CCP is the part of CP/M that allows programs to be loaded into the TPA and run. It also permits files to be created and deleted, and performs other "housekeeping" functions. The CCP is discussed in more detail in the section on built-in commands in this chapter.

Note

The "command level" mode of operation is when CP/M controls the computer and all commands are executed through the CCP. The inverse is the "program level" mode of operation, when a program controls the computer and permits only certain commands to be typed.

The *BIOS* (Basic Input and Output System) is the software module which contains the assembly language routines in CP/M that are *machine-dependent*. These are the routines that are written for a specific implementation of CP/M (in this case, the Premium SoftCard IIe System in the Apple IIe computer). The BIOS module contains all the I/O programs for communicating with the terminal, the disk controller interface, and other I/O devices.

The *BDOS* (Basic Disk Operating System) is the software module which manages the disk subsystem. The BDOS, unlike the BIOS, is *machine-independent*. The assembly language routines contained in the BDOS module are the same for all computers, regardless of the disk drive interface circuitry or the particular combination of I/O devices connected to the computer (the system configuration). The BDOS can be considered the core, or the heart, of CP/M.

Because the BDOS and the CCP modules are generally the same for all computers and the BIOS can be modified for each type of computer, CP/M can run on a variety of computers.

I/O Communication

CP/M communicates with I/O devices through the BIOS module. The BIOS module contains four 2-part interface routines that can be modified to accommodate a wide variety of I/O devices. The four I/O interface routines are each divided into two categories: *logical devices* and *physical devices.*

The logical device is an assembly language subroutine in the software that is a logical representation of the I/O function (as opposed to an actual device). In CP/M there are four logical devices, each corresponding to a general I/O function. They are: CONSOLE (CON:), for input and output to and from a console or terminal; READER (RDR:) for input; PUNCH (PUN:) for output only; and LIST (LST:) for output to a listing device, such as a printer.

The other part of the I/O interface is the physical device. The physical device is a vector that points to an assembly language routine called a *driver.* (A vector is an address containing an instruction that causes the CPU to "jump" to another address that is usually the start of another routine.) A driver routine is the software that is written to communicate with a specific type of physical device.

CP/M Physical Devices

In the SoftCard implementation of CP/M there are 11 physical devices; each corresponds to a specific type of I/O device. The following list shows the possible physical devices for the SoftCard version of CP/M.

Device	Description
TTY:	(Teletype) Normally points to the Apple keyboard and monitor.
CRT:	(Cathode ray tube) Same as TTY:, but used for an external terminal.
UC1:	(User-defined console device) An I/O device that can be used for input or output.

Device	Description
PTR:	(Paper-tape reader) Used for an input-only device from slot 2.
UR1:	(User-defined reader #1) Same as PTR:, but can be modified by the user.
UR2:	(User-defined reader #2) This device is the same as UR1:.
PTP:	(Paper-tape punch) Any standard Apple interface board capable of processing output from accessory slot 2.
UP1:	(User-defined punch #1) Same as PTP:, but can be modified by the user.
UP2:	(User-defined punch #2) This device is the same as UP1:.
LPT:	(Line printer) The LPT: device is any standard Apple interface board installed into slot 1 capable of receiving output.
UL1:	(User-defined list device) Same as LPT:, but can be modified by the user.

The Apple IIe computer communicates with I/O devices using a method called *memory-mapped I/O*, so that each of the physical device routines communicates with a specific accessory slot. Most driver subroutines are located in the ROM of the installed interface board in the accessory slot.

An example of how memory-mapped devices work would be a line printer. The interface board for the line printer must be installed in slot 1. For CP/M to communicate with the line printer, the LPT: physical device points to the slot address of accessory slot 1. The actual communication subroutine for communication between the LST: logical device and the printer is contained in ROM on the interface board.

One advantage of using the logical device/physical device interface is that it permits you to select the I/O device you wish to communicate with, by using an operating system command or a statement in a program. For example, if you are using two printers, printer output can be directed to the printer of your choice by using the STAT program. (STAT changes logical to physical device assignments.)

Logical to Physical Device Assignments

The possible logical to physical device assignments are noted below. (The first physical device for each logical device is the normal assignment.)

Logical Device	Valid Physical Devices
CON:	TTY:, CRT:, UC1:, BAT:
RDR:	TTY:, PTR:, UR1:, UR2:
PUN:	TTY:, PTP:, UP1:, UP2:
LST:	LPT:, TTY:, CRT:, UL1:

Note

BAT: (Batch processing mode) directs input from the currently assigned RDR: device to the CON: device. Output is directed to the currently assigned LST: device.

The other advantage of the logical/physical device system is that it can be used in many different implementations, thus freeing the programmer from adding any additional code for each system's particular I/O configuration. The user doesn't need to monitor which I/O devices are connected to the computer and in what particular configuration.

Communication with the disk drive system is similar to I/O communication, but is confined to the logical disk device (DSK:) being assigned to the available disk drives. Only one disk drive (of the four possible disk drives) can be accessed at a time.

CP/M Disk Drive System

The disk drive system is the permanent storage media for CP/M. To use the disk drive system effectively, you should be familiar with how CP/M stores data on the disk drive system.

Disk Organization

Information is stored on a disk in blocks of 128 bytes. Each of these blocks is referred to as a "sector." Each sector has a unique address or location on the disk and information is stored and retrieved by telling the disk drive to read or write information to a specific sector. The sectors are laid out on the disk in concentric, circular tracks, as shown in Figure 4.4.

In the SoftCard version of CP/M, there are 16 sectors per track and 35 tracks per disk. Fortunately, when you type a command to access information on a disk, you don't have to know the track and the sector. You need to know only the name of the file and the disk that it is located on. CP/M will find the sector and track automatically.

Note

See the section "CP/M Disk Files" in this chapter for an explanation of the CP/M disk file system and instructions on how to access information on the disk.

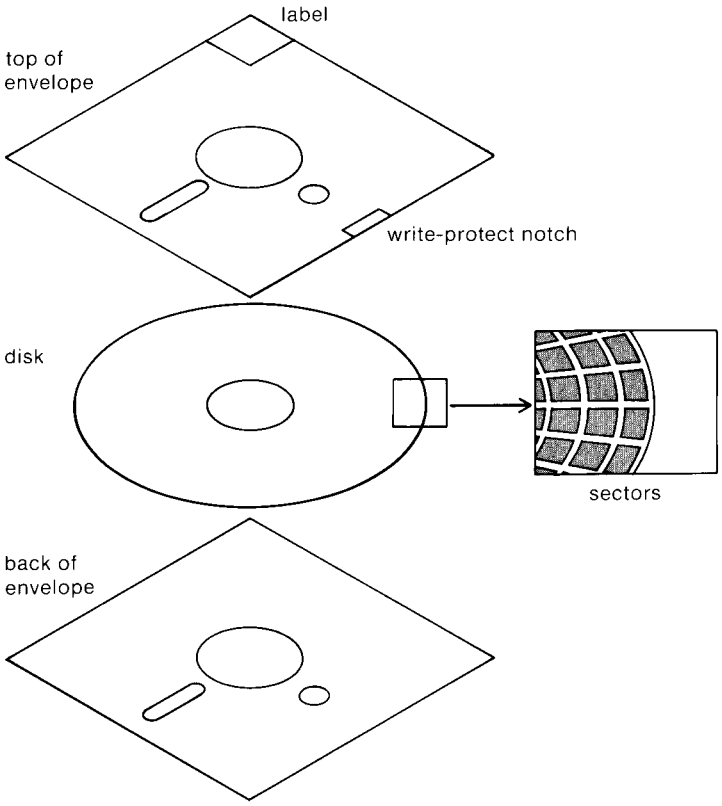


Figure 4.4. Disk Organization

CP/M Disk Types

CP/M recognizes two types of disks: system disks and data disks.

A *CP/M system disk* contains the CP/M operating system and can be loaded into memory with a warm start or a cold start from drive A:. A system disk must contain the CCP, BDOS, and BIOS modules. Any CP/M transient programs contained on the disk are optional. CP/M system disks can be created with the /S switch of the COPY utility program.

Another type of system disk is the startup disk (also called a boot disk). Startup disks are used by application programs; they have either an operating system on them or the parts of an operating system necessary to handle the program's needs. You can create your own CP/M startup disks with the AUTO-RUN program described in Chapter 6.

Data disks have no operating system data on them. They are used for the storage of programs and data files only. Since operating system information is not included, data disks have an additional 12K bytes of disk space available. Data disks are created with the /D switch of the COPY utility program.

Important

You should avoid using data disks in drive A: and in single-drive systems. The lack of an operating system on a data disk makes CP/M unable to perform a warm start and recover from errors.

Changing Disks

Whenever you change CP/M disks, you must perform a warm start because specific disk directory information is stored in memory at all times. This information is used to allocate space on the disk. When you change disks, this information must be replaced with the directory information of the newly inserted disk.

CP/M Disk Files

All data stored on disks is organized into files. A file is any collection of data, text, or program instructions. All files are referenced by file specifications, which keep track of where information is stored on the disk and what type of information it is. A CP/M file specification, or *filespec*, consists of three parts and is shown in the following notation:

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

The *d*: is the disk drive identifier, the *filename* is the filename, and .*ext* is the filename extension. All three parts are explained in the following paragraphs.

Disk Drive Identifier

The disk drive identifier is a one-letter code (A-D) followed by a colon (:). It tells CP/M which drive the file is located in. Note that the disk drive identifier is optional. If it is not included, CP/M looks for the specified file on the default or *active drive*. The active drive is the disk drive that you are currently working from. For example, when you see the A> prompt, drive A: is the disk drive that will be accessed when you give a command without a disk drive identifier. It also means that you are at the CP/M command level of operation.

Note

In other documentation, the active drive is also referred to as the *currently logged drive*.

Filename

The filename identifies the disk file and is the only required part of a filespec. CP/M filenames must start with a letter. (In some operating systems, filenames can begin with numbers or special characters.) CP/M filenames can be from one to eight characters in length and can consist of uppercase and lowercase characters. Filenames that contain both uppercase and lowercase characters will have all lowercase characters transposed into uppercase characters. For example, the filename "Program" would be transposed by CP/M into "PROGRAM."

If you include a filename extension (a three-character code) with the filename, it must be separated by a period (called a *delimiter*).

The following are examples of valid CP/M filenames:

A:MAILLIST

Refers to the file MAILLIST on drive A:.

R

Refers to the file R on the currently logged drive. Notice that this filename has only one letter.

B:BARBARA

Refers to the file BARBARA on drive B:.

Filename Extension

The filename extension denotes either the internal format of a file (the type of information in the file) or the different versions of a file. The filename extension can be from 1 to 3 characters long. For example, FNAME.1 could be the first version of the program FNAME. If you create a second version (or revise the first), you can save both versions by giving them different filename extensions (FNAME.1 and FNAME.2).

Several file types have meanings that are unique—to the CP/M operating system, to the standard CP/M transient programs, and to the high-level languages. For example, a .COM file is a "Command" file; that is, a directly executable transient program. Since certain program file operations could destroy the contents of a data file, it is a good idea to use the file type as the filename extension when you create the file. This avoids confusion when you want to use the file at a later date. Table 4.1 lists the file types commonly used for CP/M.

Table 4.1

CP/M File Types

File Extension	Type of File
.ASM	Assembly language source code
.BAK	Backup file
.BAS	BASIC source code
.COB	COBOL source code
.COM	Command file
.CRF	Relocatable assembler cross-reference
.DAT	ASCII data (FORTRAN default)
.DOC	Text document file
.FOR	FORTRAN source code
.HEX	Intel HEX format object code file
.LIB	Library file
.MAC	Macro assembler source file
.OBJ	Machine code (object file)
.OVR	COBOL compiler overlay
.PRN	Assembly language list file (PRINT file)
.REL	Relocatable object file
.SUB	SUBMIT command file
.SYM	Assembler symbol table
.TXT	Text file
.XRF	Assembler cross-reference table
.$$$	ED or PIP temporary file

Wild Card Filename Specifications

File specifications can also refer to more than one file at a time. This is done with "wild card" characters. CP/M has two wild card characters for use with filespecs: the asterisk (*) and the question mark (?). The asterisk character will match any string of characters in the filespec. The question mark character will match any character in the position occupied by the question mark during a directory search for the filename match. The following examples show some of the ways you can use wild card characters.

A:*.COM

Refers to all files on drive A: with an extension of .COM.

B:*.*

Refers to all files on drive B:.

B:????????.???

Exactly the same as B:*.* above.

DUMP.*

Refers to all files on the currently logged drive beginning with the filename "DUMP."

C*.*

Refers to any file on the currently logged drive beginning with the letter C and containing any extension.

O.

This is the same as *.*. The asterisk (*) is an abbreviation for a string of question marks (?). If an asterisk is included as part of the string, CP/M ignores all characters to the right of the asterisk and treats the whole string as a wild card character. Note the difference between this example and the next example.

?O??????.*

Refers to all files with O as the second letter of the filename on the active drive with any filename extension.

Built-in Commands

CP/M executes two types of commands, built-in and transient. Built-in commands are programs that reside permanently in the CCP module and can be used at any time. Transient commands are programs stored on a disk. Transient commands are also called transient programs.

Built-in commands are direct commands to the CPU given at the CP/M command level. They are always present whenever CP/M is active and no other programs are running. Built-in commands perform tasks such as displaying the contents of a file or a directory of disk files, renaming and erasing files, and saving the contents of memory on disk.

CP/M Built-in Commands

The SoftCard version of CP/M has seven built-in commands. Each command and its purpose is listed below. For instructions on using each of the commands, see the appropriate section in Chapter 6.

Command	Purpose
d:	Logs onto another disk drive
DIR	Displays a directory of files on disk
ERA	Erases a file or files
REN	Renames a file

Command	Purpose
SAVE	Saves the contents of memory in a file on disk
TYPE	Displays the contents of a file on the monitor screen
USER	Creates another area within the same directory

CP/M Transient Programs

A transient program is a program that can be executed as a command. The main difference between built-in commands and transient programs is that transient programs are stored on disk until they are executed. Built-in commands are stored in the CCP module in memory. Transient programs perform operations associated with programming and utility tasks such as copying files and transferring data between devices.

When not in use, transient programs are stored on disk in .COM (command) files. When you type the name of a .COM file, CP/M will load the contents of the file into memory and execute the instructions it finds in the file.

Most CP/M commands and transient programs (with a few exceptions, such as REN) are *extensible*. That is, they may be extended semantically to include additional operations. For example, the DIR command could include an argument (an entry you type in the command line) for a list of specific file types (such as BASIC files). In this case, you could type **.BAS* in the DIR command line. This instructs CP/M to display only those files with the extension of .BAS (BASIC files).

SoftCard CP/M includes 16 transient programs. Table 4.2 lists the names of the programs, their purpose, and the section of the manual that gives instructions on their use.

Table 4.2
CP/M Transient Programs

Program	Purpose	Refer to:
APDOS	Transfers text files and binary files from Apple DOS to CP/M.	Chapter 6
ASM	* Assembles 8080 assembly language programs.	*Osborne CP/M User Guide*
AUTORUN	Automatically executes a previously specified CP/M command line when the system is booted.	Chapter 6
BOOT	Exits CP/M and reboots your Apple IIe system.	Chapter 6
CAT	Displays an alphabetical listing of the directory on the specified drive.	Chapter 6
COPY	Makes duplicate copies of disks. Options to COPY let you format disks and create CP/M system disks.	Chapter 6
DDT	* Debugs 8080 assembly language programs.	*Osborne CP/M User Guide*
DUMP	* Displays a file in hexadecimal form.	*Osborne CP/M User Guide*
ED	* Creates and edits CP/M text files.	*Osborne CP/M User Guide*
LOAD	* Converts a .HEX file into a .COM file.	*Osborne CP/M User Guide*
MFT	Copies files from one disk to another on a single-drive system.	Chapter 6

Program	Purpose	Refer to:
PATCH	Makes program updates and modifications.	Chapter 6
PIP	* Copies and/or appends disk files and devices.	Chapter 6
STAT	* Displays status and makes device assignments.	Chapter 6
SUBMIT	* Batch processes commands from a disk file.	*Osborne CP/M User Guide*
XSUB	* Allows character input to a program from a Submit input file.	*Osborne CP/M User Guide*

* These programs are part of CP/M 2.2 and were written by Digital Research, Inc. All other transient programs were written by Microsoft.

Chapter 5
Using CP/M With the Apple IIe Computer

This chapter explains how CP/M works with the Premium SoftCard IIe System and the Apple IIe computer.

Using the Apple IIe Keyboard With CP/M

With CP/M, the typewriter portion of the keyboard (see Figure 5.1) works as it does with the Apple DOS or Apple Pascal operating systems. Several keys, however, work differently and there are certain CONTROL key sequences that are unique to CP/M.

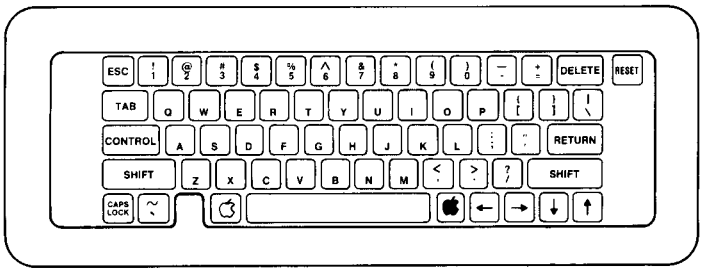


Figure 5.1. Apple IIe Keyboard

Keys You Must Use Precisely

The keys described in the "Keys You Must Use Precisely" section of the *Apple IIe Owner's Manual* must be used exactly as described in that section. The only exception is the DELETE key. The DELETE key has been redefined by the Premium SoftCard IIe System as the CP/M key RUBOUT.

Cursor Movement Keys

The left cursor key (←) and the DELETE key delete characters as they move over them. The TAB key moves the cursor seven spaces to the right. The TAB and RETURN keys are used in the same manner with CP/M as with Apple DOS and Apple Pascal.

Unless special software is provided by an application program, CP/M does not support cursor movement with the right, up, or down cursor keys.

Apple Escape Key Sequences

CP/M does not support Apple DOS ESC key sequences for cursor movement or editing. The SoftCard IIe version of CP/M does, however, support two ESC key sequences. These are ESC-(and ESC-) and they are part of the default screen function interface. Pressing the ESC-(keys and then RETURN switches the screen display to inverse video (dark characters on a light background.) Pressing ESC-) and RETURN switches the screen back to the normal display.

Apple IIe Special Function Keys

The OPEN-APPLE and SOLID-APPLE keys are used as described in the *Apple IIe Owner's Manual*. That is, pressing the OPEN-APPLE key has the same effect as pressing the button on hand-control 0; and pressing the CLOSED-APPLE key has the same effect as pressing the button on hand-control 1.

Pressing the RESET key while you are at CP/M command level will not have any effect. Pressing the CONTROL-RESET keys will cause a CP/M cold start.

Line Editing Commands

CP/M supports several line editing commands that allow you to edit a CP/M command line or to edit data input to CP/M transient programs. Most line editing commands are executed by using CONTROL characters. CONTROL characters (denoted by "CONTROL-") are used by first pressing the CONTROL key and then holding it down while you type the indicated character. Do not press RETURN after typing a CONTROL character. Table 5.1 lists the CONTROL characters associated with line editing commands.

Table 5.1.

Line Editing Commands

Key	Function
←	Moves the cursor one character position to the left and deletes characters as the cursor passes over them.
↓	Backspaces and deletes the entire line.
CONTROL-E	Moves the cursor to the beginning of the next line. However, the previous line is not terminated until the RETURN key is pressed. Note that the carriage return/linefeed character sequence generated by CONTROL-E is not entered into a line, but only sent to the console.
CONTROL-H	Same as the left cursor key (←).
CONTROL-J	Terminates input (linefeed).
CONTROL-M	Functions the same as RETURN.
CONTROL-P	Sends all ASCII character output to the printer and to the monitor screen. This "printer echo" mode remains in effect until CONTROL-P is typed, or until a CP/M warm start is performed. CONTROL-P is accepted only when console input is required.
CONTROL-R	Redisplays the current line.
CONTROL-S	Suspends ASCII character output to the terminal. Output resumes when any other key is pressed.
CONTROL-X	Same as the down cursor key.
DELETE	Same as the left cursor key.

Type-ahead Buffer

The SoftCard IIe version of CP/M has a type-ahead buffer for keyboard input. This permits you to enter commands and text while CP/M is performing other operations. When CP/M finishes an operation, it scans the type-ahead buffer for additional commands and data. This ensures that none of your input from the keyboard is lost. The type-ahead buffer holds up to 256 characters.

Using I/O Devices With CP/M

The default I/O device assignments for the SoftCard IIe version of CP/M are as follows:

Logical Device	Physical Device
CON:	TTY: (Apple keyboard and monitor)
RDR:	TTY: (Apple keyboard and monitor)
PUN:	TTY: (Apple keyboard and monitor)
LST:	LPT: (interface board in accessory slot 1)

The TTY: physical device communicates with the Apple keyboard and screen monitor if there is no interface board installed in accessory slot 3. If there is an interface board installed in slot 3, the TTY: physical device will communicate with the device connected to the installed board.

The LST: physical device communicates with the interface board installed in accessory slot 1. If there is no interface board installed in slot 1, the TTY: physical device is used.

To use other physical devices, special I/O software must be added to the patch areas of CP/M to define the location of the physical device (i.e., the accessory slot the physical device will communicate with). See "I/O Configuration" in the *Premium SoftCard System IIe Programmer's Manual* for instructions on adding I/O software to the patch areas.

Print Operations

CP/M provides several methods for sending data to a printer. The printer, however, must be connected to an appropriate interface board in the recommended accessory slot. See Table 2.1 in Chapter 2 for a list of recommended slot assignments.

Once all physical connections are made, you must make sure that the logical LST: device is assigned to the right physical device. Check the assignments with the STAT command by typing

```
STAT DEV:
```

and pressing the RETURN key. If the LST: logical device is not assigned to the LPT: or UL1: physical device, then use STAT to change the device assignment. See "I/O Communication" in Chapter 4 for an explanation of logical and physical devices. Instructions for using STAT are given in the "STAT" section in Chapter 6.

Note

The LPT: physical device is the I/O interface board in accessory slot 1. UL1: is an undefined I/O interface board.

When the LST: logical device assignment has been made, you can use either CP/M commands or the commands provided by the application program to send output to the printer.

Many application programs, such as text editors and electronic spreadsheets, have built-in print functions. The ability to send data to the printer is included as part of the program. If your program does have commands or statements for sending data or files to a printer, you should use those commands when running the application program. The major advantage of using a built-in printing program is that it usually prints your file in the format required by the application program.

If you want to print a file at CP/M command level, you can use either CONTROL-P and the TYPE command or the PIP program. Instructions on how to use TYPE and PIP are given in Chapter 6.

Running Application Programs

Most application programs have explicit instructions for loading and running the program. Others do not. If the application program you plan to use is vague about how to load the software into the Apple IIe/SoftCard system, you can follow these guidelines. If the program still can't be run, contact the computer dealer or the program manufacturer.

Guidelines for Running Application Programs

If the program is written in BASIC, it will often require the GBASIC.COM file to be loaded into memory prior to running the program.

Programs written in other languages, such as FORTRAN or COBOL, will require the appropriate language compiler or interpreter to run, unless the program is compiled into an executable (.COM) file.

Check to see how much memory the application program needs to run. The Premium SoftCard IIe System has up to 59K bytes of memory for application programs to use. If the program requires more than that, contact your computer dealer or the program manufacturer.

Some programs will require you to re-configure the screen function interface. This is an area of CP/M that translates what the program sends to the operating system into what you actually see on the terminal. See "Screen Function Interface" in Chapter 7 for more information on how to configure the interface for your program.

Application programs that are in a different disk format will require that the disk be copied to the 5 1/4-inch disk format used by the SoftCard IIe System. Contact your computer dealer for more information.

Programs created under CP/M 2.0 and earlier versions are compatible with the SoftCard IIe version of CP/M.

Important

Programs written for the previous SoftCard versions of CP/M which access specific 6502 memory addresses must be changed. This includes programs that were written under Microsoft FORTRAN-80 and Microsoft BASIC Compiler. For more information about using FORTRAN and BASIC Compiler programs with the SoftCard IIe System, see the *Premium SoftCard IIe System Programmer's Manual.*

Differences Between the Premium SoftCard IIe and Previous SoftCards

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

The following features are unique to the 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 the previous SoftCard circuit boards. The Z80B is not synchronized or phase-locked to the Apple IIe internal clocks.

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

There are no switches on the 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 SoftCard IIe circuit board, there is no need to change the size of CP/M. Therefore, the SoftCard IIe disk 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 file is not included on the SoftCard IIe Master disk. The DOWNLOAD program is on the Premium SoftCard IIe Supplement disk that comes with the *Premium SoftCard IIe System Programmer's Manual.*

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 different method of accessing the I/O system than previous versions of SoftCard CP/M. The SoftCard IIe 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 60K SoftCard accesses I/O functions through memory-mapped locations in the Apple's memory and does not use the 6502 microprocessor except to call 6502 subroutines.

The 6502 routines are called differently. 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 7-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 7-byte area.

Chapter 6
CP/M Commands and Utility Programs

This chapter explains the CP/M built-in commands and utility programs you will use most often. The commands and programs that are discussed in this chapter are listed below.

CP/M Built-in Commands

d:	Logs onto another disk drive.
DIR	Displays a directory of the files on disk.
ERA	Erases a file or files.
REN	Renames a file.
SAVE	Saves the contents of memory in a file on disk.
TYPE	Displays the contents of a file on the monitor screen.
USER	Creates another area within the same directory.

Utility Programs

APDOS	Copies text and binary files from Apple DOS disks to CP/M disks.
AUTORUN	Automatically executes a CP/M command line when the system is booted.
BOOT	Exits CP/M and reboots your Apple IIe system.
CAT	Displays an alphabetical directory listing of the drive specified.
COPY	Makes duplicate copies of disks. Options to COPY let you format disks and create CP/M system disks.
MFT	Copies files from one disk to another on a single-drive system.

PATCH	Makes program updates and modifications to CP/M.
PIP	Transfers, copies, and/or appends disk files and devices.
STAT	Displays status information and assigns devices.

Command and Program Execution

All commands and programs are executed either from CP/M *command level* or from the *program level.* The following paragraphs define both terms.

"Command level" is the CP/M command level of operation; all commands are executed through the CCP module. (See "How CP/M Uses Memory" in Chapter 4.) Command level is indicated by the CP/M command prompt (the active disk drive letter followed by the > sign).

To execute utility programs at command level, type the command line using the program name and arguments. Do not type the .COM extension. Programs executed at command level will always return control of the computer back to command level. This is useful for a single task, such as copying a single file from one disk to another.

The "program level" of operation is when a program controls the computer and permits only certain commands to be typed from the keyboard. For programs that are part of the Premium SoftCard IIe System, the asterisk prompt (*) indicates that the computer is at program level. Other application or utility programs may display another character or a *menu* to indicate program level operation.

To execute commands at program level, you must first type the name of the program and then press RETURN. The program is loaded into memory and the asterisk prompt is displayed. You may then enter the command line without typing the program name. Commands executed at the program level will return control back to the program level. This is useful for repetitive tasks, such as copying more than one disk.

Using Utility Programs

Premium SoftCard IIe utility programs can be executed at CP/M command level or at program level. The AUTORUN, BOOT, CAT, and STAT utility programs can be executed from CP/M command level only. The APDOS, COPY, MFT, PATCH, and PIP utility programs are run from CP/M command level or program level.

To stop or abort a utility program, press CONTROL-C. You can also use the line editing commands described in Chapter 5 to edit utility program commands.

Using CP/M Built-in Commands

CP/M built-in commands are executed from the CP/M command level only. Their use does not require that a system disk be in the active drive. However, if you encounter a "BDOS ERR ON d:" type of error, you must insert a CP/M system disk in the active drive to recover from the error and to continue.

Built-in Commands

The following section explains how to use the CP/M built-in commands. Examples are included with each command.

d:

The d: command allows you to change active drives in multiple-drive systems. The active drive is the disk drive that contains the CP/M system disk you are currently working from. (See "CP/M Disk Files" in Chapter 4.)

To change the active drive, type the letter which represents the drive you wish to designate followed by a colon (:), and press RETURN. For example,

```
B:
```

followed by pressing the RETURN key changes the active drive to drive B:. If you change the active drive, CP/M changes the prompt letter accordingly.

DIR

The DIR (DIRECTORY) command scans a specified disk to determine what files are on that disk. Typing *DIR* (with no arguments) displays only the sequential list of filenames on a disk in the specified drive. DIR can also display specified files when you use arguments in the command line.

Displaying a Disk Directory

DIR, when used in the following format, scans the disk directory of the disk in the drive specified and displays the directory entries (files) it finds.

```
DIR [d:]
```

Typing DIR without the *d*: argument scans the disk directory of the active drive. For example,

```
DIR
```

displays the disk directory of files on the active drive. To scan the disk directory of another drive, type the drive letter (*d*:) in the command line.

```
DIR B:
```

displays the disk directory of files on the disk in drive B:. If no files are found, CP/M displays the message:

```
NO FILE
```

Displaying Single and Multiple Disk Directory Entries

To find and display a specific file(s) on a disk, type DIR in the following format,

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

and press the RETURN key. The *d*: argument permits you to search other disk drives for the specified file or files. For example,

```
DIR B:GBASIC.COM
```

scans drive B: for the file GBASIC.COM.

To find a particular file, type the *filename.ext* of the desired file. If you want to display a certain type of file, type "wild card" characters (? or *) in the *filename.ext* argument. (Wild card characters are explained in "CP/M Disk Files" in Chapter 4.) Wild card characters also allow you to search for files that begin with a certain letter or share a common name. For example,

```
DIR D???.BAS
```

searches the disk directory of the active drive for all files beginning with D and having between one and four characters

in the filename with a filename extension of .BAS. Another way to use a wild card character in the filespec would be to type:

```
DIR *.COM
```

The *.COM instructs DIR to scan the active drive for all files with a .COM filename extension.

ERA

The ERA (ERASE) command erases specified files from a disk. You may use ERA to erase files from any disk as long as you include the file specification (filespec). Wild card characters can be used in the filespec.

Erasing a File

To erase a single file, type ERA in the following format,

ERA *filespec*

and press the RETURN key. The *filespec* is the name and the location of the file. For example:

```
ERA B:TEMP.OLD
```

erases the file TEMP.OLD from the disk in drive B:.

Erasing Multiple Files

To erase multiple files, include wild card characters (? or *) in the *filespec* argument of the command line. For example,

```
ERA C:*.BAS
```

erases all files with the extension .BAS from the disk in drive C:. Another example would be:

```
ERA *.*
```

This command line erases all files on the disk in the active drive. If you attempt to erase all the files on a disk, CP/M will ask: ALL (Y/N)? If you want to erase all the files on the disk, press the *Y* key. Otherwise, press either the *N* or the RETURN key.

REN

The REN (RENAME) command renames files while leaving the file text intact. Unlike the other built-in commands, wild card characters cannot be used in the filespec. Therefore, you can only rename one file at a time.

Renaming a File

To rename a file, type REN in the following format:

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

Press the RETURN key to execute the command. The *new filename.ext* argument is the new name of the file and *old filename.ext* is the original name of the file. For example,

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

renames the file TEMP.OLD as TEMP.NEW on the active drive.

If the file is on a disk drive other than the active drive, include the drive identifier (*d:*) in the command line, as in the following example:

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

SAVE

The SAVE command saves the contents of memory in a specified disk file. It is used mainly for program development.

Saving Memory Contents in a Disk File

You can save what you entered into memory by typing SAVE in the following format:

SAVE *nnn filespec*

The *nnn* argument is the number of memory pages to be saved. (A page of memory is equal to 256 bytes.) The *filespec* is the drive and the file where you will save the memory contents. For example,

```
SAVE 2 B:DATA.BIN
```

saves 2 pages of memory in a file called DATA.BIN on disk drive B:.

To use SAVE, you must know how many memory pages are to be saved. The memory pages to be saved will start at memory location 100H (hexadecimal). The *nnn* argument, however, must be entered as a decimal number. Instructions on how to convert the hexadecimal address to a decimal number are given in the *Osborne CP/M User Guide*.

TYPE

The TYPE command displays the contents of a specified text file on the screen. This provides a quick way of examining a file for errors or to check the contents. It can also be used to print a file in conjunction with the CONTROL-P line editing key.

Note

If you attempt to TYPE a file that is not a text file, meaningless characters will appear with unpredictable results.

Displaying a Text File on the Monitor Screen

To display a file on the monitor screen, enter TYPE in the following format:

```
TYPE filespec
```

The *filespec* is the location and name of the file. No wild card characters are allowed in the filespec. For example,

```
TYPE DUMP.ASM
```

displays the contents of the file DUMP.ASM in the active drive on the screen.

Printing a Text File With TYPE

The CONTROL-P is an on/off line editing command that controls the output to the printer. When used with the TYPE command, it permits you to print the contents of a text file while it is being displayed on the screen.

Note

CONTROL-P assumes there is a print device assigned to the LST: logical device. Before printing a file with TYPE, check the logical device assignments with the STAT command.

To print a text file, press CONTROL-P and then type the command:

TYPE *filespec*

and press RETURN. For example, press CONTROL-P, then type:

```
TYPE B:DUMP.ASM
```

Press RETURN to execute the command. This example will print the file DUMP.ASM in drive B:, as it is being displayed on the screen.

USER

The USER command separates disk memory into user areas. The user areas are designated by numbers (i.e., 0, 1, 2, and so on). This command is useful for creating multiple file directories (one per user area) on disks. However, USER is of limited value with floppy disks because of the small memory areas they contain. If you have a hard disk, USER allows you to maintain separate memory areas on the same disk.

Copying files from one user area to another is described in the "PIP" section of this chapter.

Creating a User Area

To create a new user area on the disk in the active drive, type USER in the following format,

USER *n*

and press the RETURN key. The *n* argument is the number (any unused number between 0 and 15) of the new user area.

If the specified number hasn't been used on that particular disk, USER creates the user area for that number. For example, if you already have three user areas (0, 1, 2) and you wish to create a fourth one, type

USER 3

and press RETURN. This immediately creates user area 3 and transfers you to that area.

Changing the Active User Area

To change the active user area, type

USER *n*

and press the RETURN key. The *n* argument is the number of the desired user area. For example, type

USER 0

to change the active user area back to area 0.

Note

If you attempt to execute a program in a .COM file which is not in the current user area, CP/M automatically searches user area 0 for that file. This applies to .COM (Command) files only.

Utility Programs

The following section explains the CP/M utility programs included in the Premium SoftCard IIe package. These are the programs that you will use most often. Examples are given with each program.

APDOS

APDOS.COM is a utility program that copies Apple text and data files from Apple DOS disks to CP/M system disks.

Note

Apple DOS text and data files are usually incompatible with CP/M. You can, however, copy the files to CP/M system disks and modify them with a text editor.

APDOS can copy files from CP/M command level or from the APDOS program level. From CP/M command level, the command line is typed and then executed by pressing the RETURN key. APDOS will exit to CP/M command level after each execution.

To run APDOS at the program level, type *APDOS* and press RETURN. The program level command line can then be executed when the asterisk prompt appears. APDOS will return to the program level prompt after executing the command line.

Copying Single Apple DOS Data and Text Files to CP/M

Use the following procedure to copy a single Apple DOS data or text file to CP/M. This procedure assumes that you are at CP/M command level.

1. Put a CP/M system disk with the file APDOS.COM into drive A:.

2. Type the name of the file to be transferred in the following format,

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

   where *d*: specifies the destination disk drive and *s*: specifies the source disk drive. For single-drive systems, enter *A*: for both source and destination drives.

   The *cp/mfilename.ext* argument is the name of the CP/M destination file and *dosfilename* is the name of the Apple DOS file you wish to copy. When you have typed the command line, press the RETURN key to execute the command.

3. When APDOS has been loaded into memory, it will then display:

   INSERT APPLE DOS DISK IN DRIVE *s*:
   INSERT CP/M DISK IN DRIVE *d*:
   PRESS RETURN TO BEGIN

   Follow the instructions shown on the screen and insert the disks into the appropriate drives. For single-drive systems, you will have to change the disks in drive A: after the Apple DOS file has been copied into memory.

   When the copy process has been completed, APDOS displays the message

   TRANSFER COMPLETE

   and exits to CP/M command level.

4. Type

   DIR *filespec*

   and press the RETURN key to verify that the Apple DOS file has been transferred to your CP/M disk.

Copying Multiple Apple DOS Data and Text Files to CP/M

Use the following procedure to copy multiple Apple DOS data and text files to CP/M. The procedure assumes that you are starting at CP/M command level.

1. Put a CP/M system disk with the file APDOS.COM into drive A:.

2. Type *APDOS* and press RETURN. When you see the asterisk prompt, type the name of the file to be transferred in the following format:

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

   cp/mfilename.ext is the name of the CP/M destination file and *dosfilename* is the name of the Apple DOS file you want to copy. When you have typed the command line, press the RETURN key to execute the command.

3. When APDOS is loaded into memory, it will display:

   INSERT APPLE DOS DISK IN DRIVE *s*:
   INSERT CP/M DISK IN DRIVE *d*:
   PRESS RETURN TO BEGIN

   Follow the instructions shown on the screen and insert the disks into the appropriate drives. For single-drive systems, you will have to change the disks in drive A: during the copy process. The screen will display instructions for doing so.

   When the copy process has been completed, APDOS displays the message

   TRANSFER COMPLETE

   and displays the APDOS program prompt. Repeat steps 2 and 3 for each Apple DOS file you plan to copy.

4. To exit APDOS, press CONTROL-C.

5. Type

   DIR *filespec*

   and press the RETURN key to verify that the Apple DOS file has been transferred to your CP/M disk.

Copying Apple BASIC Programs to CP/M

The format of Apple DOS BASIC files (Applesoft BASIC or Integer BASIC) must be modified before they can be copied with the APDOS program. Use the following procedure to modify and copy Apple DOS BASIC files to a CP/M disk.

1. Insert the source Apple DOS disk into drive A:.

2. Load Apple DOS into memory by pressing the CONTROL and RESET keys or by turning on the power. When the Apple DOS prompt (*) appears, type

   LOAD *filename*

   and press the RETURN key to execute the command.

3. Type *LIST* to obtain a listing of the file. After the listing appears, type the following line as the first line of the program:

```
0 PRINT"CHR$(4)+"OPEN APPLEPROG":PRINT"CHR$(4)
WRITE APPLEPROG":POKE 33,33:LIST:PRINT
"CHR$(4) CLOSE":END
```

Note

The program line shown above should be typed without pressing the RETURN key. It may appear to be two lines as you type it, but it is actually only one line in the BASIC program.

Press the RETURN key when finished.

4. Run the program by typing

   ```
   RUN
   ```

   and then pressing the RETURN key. The program makes a text file copy of your program called APPLEPROG.

5. Remove the Apple disk from drive A:. Insert a CP/M system disk containing the APDOS program into drive A:.

6. Load CP/M into memory by typing

   ```
   PR#6
   ```

   and press the RETURN key.

7. At CP/M command level, type

   ```
   APDOS
   ```

   and press the RETURN key.

8. If you have a multiple-drive system, type

   ```
   APPLE.BAS=APPLEPROG
   ```

   at the APDOS program prompt (*), and press the RETURN key. If you have a single-drive system, type

   ```
   APPLE.BAS=A:APPLEPROG
   ```

   and press the RETURN key.

9. Follow the instructions shown on the screen.

10. Exit APDOS by pressing CONTROL-C.

11. At CP/M command level, type

    ```
    GBASIC
    ```

    and press the RETURN key.

12. When you see the GBASIC "Ok" prompt, type

```
LOAD "APPLE"
```

and press the RETURN key.

13. Delete line 0 (the line entered in step 3).

This completes the APDOS BASIC file copy procedure. A copy of your Apple DOS BASIC program (called APPLE.BAS) now exists on the CP/M disk and in memory. If you want to copy more than Apple BASIC file to the same CP/M disk, rename the APPLE.BAS file and repeat the procedure.

Note

Because of the differences between Apple BASIC and Microsoft BASIC, the copied program must be edited to change the Apple POKE, PEEK, CALL, and disk file statements into their equivalent Microsoft BASIC statements. See Appendix D in the *Microsoft BASIC Interpreter Reference Manual* for more information on converting programs to Microsoft BASIC.

AUTORUN

AUTORUN.COM is a utility program that permits you to create startup disks. Startup disks are system disks that automatically execute a startup program when a cold or warm start is performed.

When you create a startup disk with AUTORUN, any CP/M command line (that is, any CP/M program or command) can be automatically executed each time the system is started with either a warm or cold start. Thus, you can automatically load and run programs without ever seeing the CP/M A> prompt.

Creating Startup Disks

Use the following procedure to create a CP/M startup disk.

1. Load CP/M into memory.

2. Use the PIP utility program to copy AUTORUN.COM onto the system disk that you plan to use.

3. At CP/M command level, type the command line in the following format,

   AUTORUN [*command line*]

   and press the RETURN key. The *command line* argument is any executable CP/M program name or CP/M built-in command. For example,

   AUTORUN CAT

   displays the directory on the default drive when you boot the system.

Repeat the procedure for each startup disk.

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

Loading Startup Disks

To execute a startup disk, it must be loaded in the active drive (usually drive A:). When you start the system, the command line will be executed immediately after the CP/M operating system modules are loaded into memory. For example,

AUTORUN GBASIC PROG

loads and runs GBASIC after CP/M. The BASIC program PROG is loaded and executed after GBASIC.

BOOT

BOOT.COM is a program that reboots your Apple IIe computer from any system disk at CP/M command level. 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.

Loading CP/M With BOOT

Insert a system disk that contains BOOT.COM into the active drive. Type

```
BOOT
```

and press the RETURN key. BOOT executes the cold start loader in ROM which loads CP/M.

Note

When BOOT is executed, the operating system is reloaded and all programs that were in memory are erased.

Loading Other Operating Systems With BOOT

To load any other operating system, insert a CP/M system disk containing BOOT.COM into the active drive. Type BOOT in the following format,

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

and press the RETURN key. The *number* argument is the slot number (4, 5, or 6) of the disk controller board connected to the disk drive you are loading from. 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.)

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

After you have pressed RETURN, the screen displays:

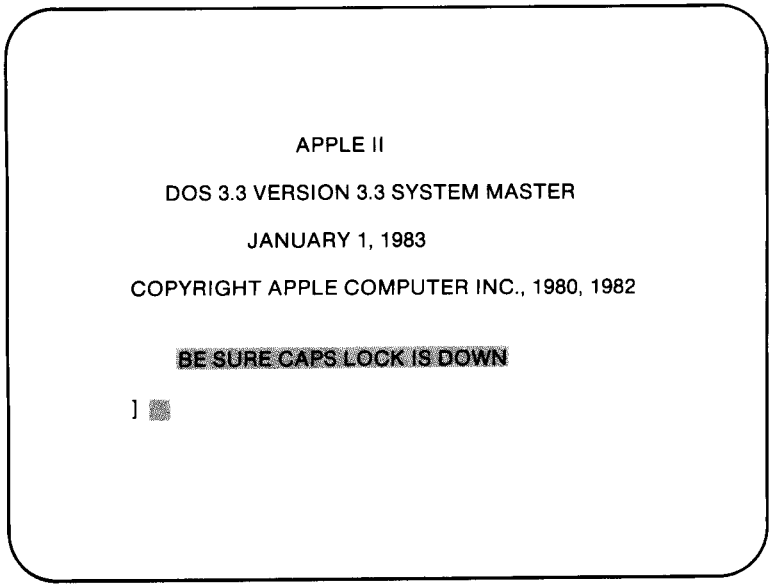
```
INSERT DISK AND PRESS RETURN TO REBOOT SYSTEM:
```

Insert the system disk into the appropriate drive. If you are loading an Apple DOS 3.2 disk, press the *3* key. If you are loading an Apple DOS 3.3 disk, press the RETURN key.

After you press the appropriate key, the operating system will be loaded into memory. This will be indicated by a logon message. For example, if you have a CP/M disk in drive A: (the active drive), and want to load Apple DOS from drive B:, type

```
BOOT 6
```

and press RETURN. When the BOOT prompt appears on the screen, press the RETURN key. Apple DOS 3.3 should be loaded into memory and the following logon message is displayed:



CAT

The CAT utility program is similar to the DIR command. It scans the directory of a disk to determine which files are on that disk. The list displayed by CAT, however, is in alphabetical order and shows the size of each file and the amount of remaining unused disk space (in kilobytes).

Displaying a Disk Directory

CAT, when used in the following format, scans the directory of a disk in the specified drive and displays all the directory entries (files) it finds.

```
CAT [d:]
```

Typing CAT without the *d*: argument scans the disk directory of the active drive. For example, to scan the SoftCard Master disk in the active drive, type

```
CAT
```

and press the RETURN key. The CAT program would display the files as follows:

```
APDOS    .COM  2K | CONFIGIO .BAS  7K | ED      .COM  7K | PIP     .COM  8K |
ASM      .COM  8K | COPY     .COM  2K | GBASIC  .COM 26K | STAT    .COM  6K |
AUTORUN  .COM  1K | DDT      .COM  5K | LOAD    .COM  2K | SUBMIT  .COM  2K |
BOOT     .COM  1K | DUMP     .COM  5K | MFT     .COM  2K | XSUB    .COM  1K |
CAT      .COM  1K | DUMP     .ASM  1K | PATCH   .COM  1K |

Total of 113K bytes in 19 files, 25K bytes available
```

Compare the result of the CAT command to that obtained by the DIR command (see the "Built-in Commands" section of this chapter).

To scan the directory of another drive, include the drive identifier (*d*:) with the CAT command. For example,

```
CAT B:
```

scans the disk directory in drive B:. If no files are found, CAT displays the message:

```
NO FILE
```

Displaying Single and Multiple Disk Directory Entries

To find and display a specific file or files on a disk, type CAT in the following format,

```
CAT filespec
```

and press the RETURN key. The *filespec* argument is the name and location of the file or files sought. For example,

```
CAT B:GBASIC.COM
```

scans drive B: for the file GBASIC.COM. If the file is found, CAT displays:

```
GBASIC.COM
Total of 26K bytes in 1 file, 25K bytes available
```

To search for a type of file, use wild card characters (? or *) in the filename and extension part of the *filespec* argument. (Wild card characters are explained in "CP/M Disk Files" in Chapter 4.) Wild card characters also allow you to search for files that begin with a certain letter or share a common name. For example,

```
CAT A???.BAS
```

searches the disk directory of the active drive for all files beginning with the letter A having between one and four characters in the filename, with a filename extension of .BAS. Another way to use a wild card character in a filename would be to type

```
CAT *.COM
```

which scans the active drive for all files with a .COM filename extension.

CAT

The CAT utility program is similar to the DIR command. It scans the directory of a disk to determine which files are on that disk. The list displayed by CAT, however, is in alphabetical order and shows the size of each file and the amount of remaining unused disk space (in kilobytes).

Displaying a Disk Directory

CAT, when used in the following format, scans the directory of a disk in the specified drive and displays all the directory entries (files) it finds.

```
CAT [d:]
```

Typing CAT without the *d*: argument scans the disk directory of the active drive. For example, to scan the SoftCard Master disk in the active drive, type

```
CAT
```

and press the RETURN key. The CAT program would display the files as follows:

```
APDOS    .COM  2K | CONFIGIO .BAS  7K | ED     .COM  7K | PIP    .COM  8K |
ASM      .COM  8K | COPY     .COM  2K | GBASIC .COM 26K | STAT   .COM  6K |
AUTORUN  .COM  1K | DDT      .COM  5K | LOAD   .COM  2K | SUBMIT .COM  2K |
BOOT     .COM  1K | DUMP     .COM  5K | MFT    .COM  2K | XSUB   .COM  1K |
CAT      .COM  1K | DUMP     .ASM  1K | PATCH  .COM  1K |

Total of 113K bytes in 19 files, 25K bytes available
```

Compare the result of the CAT command to that obtained by the DIR command (see the "Built-in Commands" section of this chapter).

To scan the directory of another drive, include the drive identifier (*d*:) with the CAT command. For example,

```
CAT B:
```

scans the disk directory in drive B:. If no files are found, CAT displays the message:

```
NO FILE
```

Displaying Single and Multiple Disk Directory Entries

To find and display a specific file or files on a disk, type CAT in the following format,

```
CAT filespec
```

and press the RETURN key. The *filespec* argument is the name and location of the file or files sought. For example,

```
CAT B:GBASIC.COM
```

scans drive B: for the file GBASIC.COM. If the file is found, CAT displays:

```
GBASIC.COM
Total of 26K bytes in 1 file, 25K bytes available
```

To search for a type of file, use wild card characters (? or *) in the filename and extension part of the *filespec* argument. (Wild card characters are explained in "CP/M Disk Files" in Chapter 4.) Wild card characters also allow you to search for files that begin with a certain letter or share a common name. For example,

```
CAT A???.BAS
```

searches the disk directory of the active drive for all files beginning with the letter A having between one and four characters in the filename, with a filename extension of .BAS. Another way to use a wild card character in a filename would be to type

```
CAT *.COM
```

which scans the active drive for all files with a .COM filename extension.

Note

When CAT is used with drives other than Apple Disk II drives (such as a hard disk), the displayed file size will differ from that shown by the STAT program. CAT displays the actual size of the file, while STAT displays the amount of space on the disk allocated by the file. The latter figure may be larger.

COPY

COPY.COM is a utility program that copies and formats CP/M disks. By using its software "switch" options, you can:

Format a disk with the /F switch.

Copy the contents of one disk onto another (no switch needed).

Verify that the copied disk contents match the contents of the original disk with the /V switch.

Create CP/M system disks with the /S switch.

Create CP/M data disks with the /D switch.

Like other utility programs, COPY can be used from CP/M command level or from the COPY program level.

Formatting a Disk

Disk formatting prepares the disk to store data in a certain format. Whenever CP/M system disks or data disks are created, they are formatted automatically. This is also true if you copy the entire contents of one disk onto another. To perform the formatting function only, for just one disk, type the COPY command line in the following format from CP/M command level:

```
COPY d:/F
```

The *d*: argument specifies the drive of the disk to be formatted. The /F is a software switch that instructs COPY to format the disk only. For example,

```
COPY B:/F
```

formats the disk in drive B:.

To format disks from the program level, type *COPY* and press RETURN. When you see the asterisk prompt, type:

```
d:/F
```

In either command, the formatting process is the same. If you plan to format several disks at a time, use COPY from the program level.

Copying CP/M Disks

The most common use of COPY is to copy the entire contents of one disk to another. To copy disks from CP/M command level, type the COPY command line in the following format:

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

Press the RETURN key to execute the command. The *d*: argument is the *destination drive* (the drive you wish to copy to, A: through D:). The *s*: argument is the *source drive* (the drive you

are copying from, A: through D:). The /V switch is the "verify copy" option. It instructs COPY to verify that no errors occurred during the copying process. For example,

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

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

To use COPY from the program level, type

```
COPY
```

and press the RETURN key to load the COPY program into memory. When the asterisk prompt (*) appears, use the same command line arguments as used at CP/M command level. Pressing the RETURN key at program level executes the command. For example, if you first type *COPY* and press RETURN, the command line

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

will perform the same copy process as in the previous example, but it will be executed from the program level. When the copy process has been completed, COPY returns to the program level.

While the COPY program is running, it will give you instructions for each step of the copy process and status messages. See the section entitled "Backing Up the SoftCard Master Disk" in Chapter 3 for examples of screen instructions and messages.

Note

You can use COPY with either a single-drive system or a multiple-drive system. With a single-drive system, you must specify the destination drive and the source drive as the same drive.

Creating CP/M System Disks

A *CP/M system disk* contains the CP/M operating system and can be loaded into memory with a warm start or a cold start from drive A:.

CP/M system disks created with the /S switch include only the CP/M operating system software with the CP/M built-in commands; they do not include CP/M utility programs. Utility programs must be copied onto a CP/M system disk with the MFT or PIP programs.

To create CP/M system disks, use COPY in the following format:

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

Press the RETURN key to execute the command. The *d*: argument is the destination drive (A: through D:). /S is a software switch that instructs COPY to copy only the CP/M operating system onto the first three tracks of the disk. The /F switch formats the disk and the /V switch verifies the copy process. For example,

```
COPY B:/S
```

copies the operating system software from the disk in drive A: onto the disk in drive B:.

To use COPY from the program level, type

```
COPY
```

and press the RETURN key to load the COPY program into memory. When the asterisk prompt (*) appears, use the same command line arguments as used at CP/M command level. Pressing the RETURN key at program level executes the command.

Note

When 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 unless the /F switch is used.

Creating CP/M Data Disks

Data disks are disks that have no operating system data on them. They are used for the storage of programs and data files only.

Important

You should 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.

To create CP/M data disks, use COPY in the following format:

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

Press RETURN to execute the command. The /D switch instructs COPY to create a data disk. As with the /S switch, if the disk has been formatted, the files on the disk will not be deleted unless the /F switch is used. If the disk is already a CP/M system disk, the CP/M system is removed, and an additional 12K bytes of disk space is made available for programs and data. The /V switch verifies the copy process.

To create CP/M data disks from the program level, type

```
COPY
```

and press the RETURN key to load the COPY program into memory. When the asterisk prompt (*) appears, enter the command line in the following format:

```
d:/D[/F][/V]
```

To execute the command line, press the RETURN key. The arguments are used in the same manner as from CP/M command level.

MFT

MFT.COM is a utility program for copying files from one disk to another. MFT is similar to the PIP program, but is designed specifically for single-drive systems. It can be run from either CP/M command level or from MFT program level.

Copying Files from CP/M Command Level

Before using MFT, make sure the file MFT.COM is on the system disk you plan to use. To copy files at CP/M command level, type MFT in the following format:

```
MFT filename.ext1[,filename.ext2...]
```

The *filename* arguments following MFT are the names of the files you want to copy. Wild card characters (? and *) can be used in the filenames or filename extensions. The copy process is started when you press the RETURN key. For example,

```
MFT *.COM
```

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

While the MFT program is running, it will display instructions on the screen for changing disks. Whenever you want to cancel the copy process, press CONTROL-C.

Important

You must have a CP/M system disk in disk drive A: before pressing CONTROL-C. CP/M initiates a "warm start" whenever CONTROL-C is typed from the keyboard. If you don't have a CP/M system disk in the disk drive, CP/M cannot restart itself and displays "A BDOS ERROR on d: Bad Sector" on the screen.

Copying Files from MFT Program Level

To use MFT at the program level, type *MFT* and press the RETURN key. When the MFT program is loaded into memory, the screen displays the MFT program banner and the asterisk prompt. At the asterisk prompt, type the MFT arguments in the same manner as you would from the CP/M command level. For example, if you load MFT and press RETURN, then type,

```
GBASIC.COM,CONFIGIO.BAS
```

MFT copies the GBASIC.COM and CONFIGIO.BAS files from the source disk to the destination disk.

PATCH

PATCH.COM is a utility program for installing program updates and modifications to the CP/M system modules.

The only time you should have to use PATCH is when you receive explicit instructions from a software vendor, such as Microsoft. If you wish to install your own modifications or updates without instructions from a vendor, do so at your own risk. General instructions for using PATCH are given in the *Premium SoftCard IIe System Programmer's Manual.*

PIP

PIP (Peripheral Interchange Program) is one of the most frequently used CP/M programs. The primary purpose of PIP is to transfer data between devices. Its most frequent use is in copying files from one disk to another, but PIP can also be used to:

Rename the destination file during the copy process

Copy files from different user areas to the active user area

Append disk files (concatenation)

Merge disk files

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

Copy data between I/O devices

The most commonly used functions are discussed in this section. PIP can also be used as an aid in program development and is discussed in Chapter 3 of the *Osborne CP/M User Guide.*

You can use PIP at CP/M command level or at the PIP program level. From either level, the command line arguments are the same. When PIP is executed from CP/M command level, it returns to CP/M command level after executing the command line. When executed from program level, it returns to the program level after executing the command line. To exit PIP from the program level, press CONTROL-C.

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."

Copying Files From Disk to Disk

The most common use of the PIP program is to copy files from one disk to another. To copy a file or files to another disk from CP/M command level, type the PIP command in the following format

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

and press the RETURN key.

To copy files from the PIP program level, type *PIP* and press RETURN. Then type:

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

In both formats, the *d*: argument is the *destination drive* (the drive you want to copy to) and the *s*: argument is the *source drive* (the drive you are copying from). The *filespec* argument is the file specification of the file or files you are copying from. There is no need to include the destination filespec unless the file is to be renamed. For example,

A:=B:ED.COM

copies the file ED.COM from drive B: to drive A: under the same filename. If you want to rename a file, use the same command line format as before, but specify a new filespec for the destination file. For example,

PIP DOG.COM=CAT.COM

copies the file CAT.COM into a new file called DOG.COM on the disk in the active drive.

You may also use wild card characters (* or ?) in the filespec arguments to copy more than one file. The command line

PIP B:=*.BAS

copies all files on the disk in the active drive with the extension of .BAS to the disk in drive B:.

Note

If you plan to copy more than one file, use PIP from the program level. This will save time and eliminate unnecessary keystrokes.

The optional [*p*] (parameter) argument modifies the copy process or permits certain conditions to be set. If you include a parameter argument, the square brackets must enclose the parameter. For most disk to disk copy operations, parameters are not needed.

If your disk system is divided into user areas, the [g] parameter can be used to copy files from a different user area into the active user area. For example,

```
PIP A:=B:ARCH.BAS[g4]
```

copies the file ARCH.BAS from user area 4 in drive B: to the active user area.

Important

You cannot have two files with the same filename on the same disk or in the same user area.

Copying Parts of a File

PIP can copy portions of a file when used with one of two parameters. PIP with the [G*string* CONTROL-Z] parameter copies from the beginning of a file to the point denoted by the *string* CONTROL-Z. The [S*string* CONTROL-Z] parameter instructs PIP to copy a file from *string* CONTROL-Z to the end of the file. The brackets *must* be included. For example,

```
PIP B:=A:BIO.TXT[SA minute passed. CONTROL-Z]
```

copies from the point "A minute passed." to the end of the BIO.TXT file in drive A: and places the text in a file with the same name in drive B:.

Appending Files

PIP can be used to append several text files to a destination file (*concatenation*). Before files can be concatenated, there must be sufficient space on the disk to copy the files into a destination file.

To concatenate files, type PIP in the following format:

```
PIP [d:]dest=[d:]source1,source2...
```

Press RETURN to execute the command. The *dest* argument is the destination file of the copy operation. The *source* arguments are the source files. Commas must separate the source file arguments. For example,

```
PIP BOOK.TXT=CHAP1.TXT,CHAP2.TXT,CHAP3.TXT
```

copies the text files CHAP1.TXT, CHAP2.TXT, and CHAP3-.TXT into the file BOOK.TXT on the active drive.

If you are concatenating text files, no parameters are needed. If you are concatenating other types of files (e.g., hex files), an H, I, or O must be included in the command line. For example,

```
PIP PLAN.HEX=P2A.HEX,P3A.HEX,P4A.HEX[H]
```

copies the three hex source files into the file PLAN.HEX. The [H] parameter denotes a hexadecimal data transfer.

Copying Files to a Printer

PIP can copy files to I/O devices. If a printer is connected to the LST: logical device, a file can be sent to the printer.

To print a file, type the following command line:

```
PIP LST:=[d:]filespec[parameter]
```

Press RETURN to execute the command. For text files, add the parameters [T8P60] to put the output data into the proper format for printing. The T8 parameter substitutes tab stops for spaces and the P60 parameter inserts form feed characters every 60 lines. For example,

```
PIP LST:=BOOK.TXT[T8P60]
```

copies the file BOOK.TXT to the LST: device, and substitutes eight spaces for each tab stop in the file and a form feed command every 60 lines as it copies BOOK.TXT to the LST: device.

Other Uses for PIP

PIP can also be used to copy data between devices, and for copying between devices and files. For instructions on using PIP for these tasks, see the *Osborne CP/M User Guide*.

STAT

STAT.COM is the CP/M utility program for displaying status information and changing device assignments. The functions STAT performs are:

Displaying disk drive status

Displaying active disk and user area status

Displaying file status

Displaying device assignments

Changing device assignments

Assigning attributes to files and disks

Each of these functions is discussed in the following paragraphs. STAT is executed from CP/M command level only.

Disk Drive Status

Use the following format to display the status and the amount of free disk space in a specified disk drive:

```
STAT [d:]
```

For example,

```
STAT A:
```

displays the amount of free disk space for the disk in drive A:.

If you type *STAT* with no arguments, STAT will display the amount of disk space remaining on the active drive and the assigned attributes. For example, if you have the SoftCard Master disk in drive A: and type *STAT*, you will see the following display:

```
A: R/W, Space:20K
```

Active Disk and User Area Status

In the previous example, STAT displayed only the disk attributes and the amount of free space remaining. To display the statistical data for each disk drive or user area, use the following command line format:

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

The DSK: argument permits you to display disk characteristics of the active disk drive, and the USR: argument displays the current and active user areas. For example,

STAT B:DSK:

displays:

```
   B:   Drive Characteristics
1120:   128 Byte Record Capacity
 140:   Kilobyte Drive Capacity
  48:   32 Byte Directory Entries
 128:   Records/Extent
   8:   Records/Block
  32:   Sectors/Track
   3:   Reserved Tracks
```

Status Information About Files

To display status information about files, type

STAT *filespec*

and press the RETURN key. The *filespec* argument is the name and location of the file or files you want status information on. To obtain status information on more than one file at a time, use wild card characters in the filespec. In either case, STAT

will display the size of the file (or files) in both bytes and records; the number of extents the file contains; the file attribute set; and the filename itself. For example,

```
STAT DUMP.BAK
```

displays the size and attributes of the DUMP.BAK file on the active drive in the following format:

```
Recs      Bytes     Ext     Acc
  33        5K       1     R/W A:DUMP.BAK
Bytes Remaining On A: 20K
```

Assigning Attributes to Files and Disks

You can use STAT to set certain conditions for accessing files or disks. For example, you can make a file or disk a read-only file (a file or disk that can be read from, but not changed).

To change the attributes of a file or disk, type

```
STAT {d:|filespec}$attribute
```

and press RETURN. The *attribute* argument assigns one of the attributes from the list below to the file or disk. For example,

```
STAT B:DOG.COM $R/O
```

assigns the Read-Only attribute ($R/O) to the file DOG.COM on drive B:.

Attribute	Explanation
$R/O	(Read Only) Prevents writing to or deleting the file.
$R/W	(Read/Write) Allows writing to and deleting the file. This attribute cancels $R/O.
$SYS	(System) Prevents the display of file when the DIR built-in command is invoked.
$DIR	(Directory) Cancels the $SYS attribute.

Assigning I/O Devices

One of the strong points of CP/M is that you can change the I/O device assignment of your system without having to remember the exact slot assignment of each I/O device. CP/M provides four logical device names that can be assigned to any number of I/O devices by using the STAT program. (For an explanation of physical and logical devices, see the section entitled "I/O Communication" in Chapter 4.) To make device assignments, type:

```
STAT log:=phy:
```

The *log*: argument is the logical device and *phy*: is the physical I/O device. For example,

```
STAT CON:=TTY:
```

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

To see the possible device assignments for your system, type

```
STAT VAL:
```

and press the RETURN key. STAT displays the possible STAT command arguments and device assignments. To see the current device assignments, type

```
STAT DEV:
```

and press the RETURN key.

Chapter 7
I/O Configuration

The SoftCard version of CP/M can be modified for use with different I/O devices and software. This chapter describes the following areas of CP/M that can be modified:

The screen function interface

The Keyboard Character Definition Table

Patch areas for I/O software

All three areas can be changed or examined with the CONFIGIO utility program.

CONFIGIO

CONFIGIO.BAS is a utility program that changes designated areas of the BIOS. CONFIGIO consists of a series of menus that allow you to perform the following functions:

Examine and modify the screen function interface for use with an external terminal

Redefine keyboard characters

Load user I/O driver software into designated user patch areas

Save changes made with CONFIGIO on a system disk

Running the CONFIGIO Program

The CONFIGIO program (CONFIGIO.BAS) is on the Premium SoftCard IIe Master disk. To run it, insert a CP/M system disk that contains CONFIGIO.BAS and GBASIC.COM into drive A:. Load CP/M with a cold start. When you see the CP/M command level prompt A>, type

```
GBASIC CONFIGIO
```

and press the RETURN key.

When CONFIGIO has been loaded into memory, the screen displays a menu, as shown below. Each selection allows you to perform the task named. To select a task, press the number key corresponding to the task you wish to perform.

```
+ + CONFIGIO SELECTION MENU + +

  1. Configure Screen Function Interface

  2. Redefine Keyboard Characters

  3. Load User I/O Driver Software

  4. Read/Write Changes Made

  Q. Quit Program

  Select - ▒
```

CONFIGIO Menu Selections

1. Configure Screen Function Interface

   This selection allows you to specify the control sequences required for an external terminal or application program to execute specific screen functions. Instructions for configuring the screen function interface for an external terminal are provided in the "Configuring the Screen Interface" section of this chapter.

2. Redefine Keyboard Characters

   This selection allows you to redefine the ASCII value assigned to any particular key on the keyboard, such as a seldom-used CONTROL character. Instructions for redefining keyboard characters are given in the "Redefining Keyboard Characters" section of this chapter.

3. Load User I/O Driver Software

   This option allows you to load the necessary I/O driver software into the patch areas for use with nonstandard Apple I/O devices or I/O software. If you are adding an I/O device that requires special I/O software, the technical manual for that device should give explicit instructions on how to load the I/O software into memory. If it does not, contact the manufacturer of the I/O device.

   If you are planning to add your own I/O software to the patch areas, read the *Premium SoftCard IIe System Programmer's Manual* before proceeding.

4. Read/Write Changes Made

   This option allows you to save the changes made with CONFIGIO menu selections 1 through 3. Instructions for using menu selection 4 are listed with instructions on using the other menu selections in this chapter.

Q. Quit Program

   Pressing *Q* exits the CONFIGIO program and returns to the CP/M operating system.

Screen Function Interface

The screen function interface controls how characters are displayed on the Apple screen or on the screen of an external terminal. Screen functions (also called screen attributes) are special control sequences that govern the display characteristics of the screen monitor or terminal. Some application programs are written for more than one computer and must be modified to display characters on the screen correctly.

Most popular terminals, including the standard Apple screen monitor, support special screen functions such as: direct cursor addressing, screen clear, and highlighted text. Many CP/M application programs, such as word processing packages and business software, use these functions as part of the application display. The character sequences, however, often differ from terminal to terminal.

The screen function interface is configured for the standard Apple screen monitor. The Soroc IQ™ 120/140, Hazeltine™ 1500/1510, and Datamedia terminals can be used as external terminals without any modifications to the screen function interface. If you use an external terminal that is not compatible with your application software, special assembly language subroutines must be written to resolve the differences.

Screen Function Tables

The screen function interface solves the compatibility problem by translating the functions (as they are received from the user software) into the corresponding functions expected by the screen display's circuits. This is carried out by two translation tables: the Software Screen Function Table and the Hardware Screen Function Table.

The Software Screen Function Table recognizes an incoming screen function sequence and translates it into the corresponding sequence found in the Hardware Screen Function Table. This sequence is then sent to the terminal device.

Screen Functions Supported

The screen function interface recognizes and translates the following screen functions:

Clear Screen

Clears the entire screen, fills the screen with spaces, and places the cursor in the home position.

Clear to End-of-Page

Clears all information from the cursor (including the cursor position) to the end of the page.

Clear to End-of-Line

Clears all information from the cursor (including the cursor position) to the end of the line.

Set Normal (lowlight) Text Mode

Sets the normal video display mode; characters are displayed as white characters on a black background.

Set Inverse (highlight) Text Mode

Sets the inverse video display mode; characters are displayed as black characters on a white background.

Home Cursor

Moves the cursor to the first character position on the first line.

Address Cursor

Sets the cursor address for a specified printer offset.

Move Cursor Up

Moves the cursor up one line. If the cursor reaches the top line of the screen, it remains there and no scrolling occurs.

Move Cursor Forward

Moves the cursor one cursor position to the right, but does not destroy the character in that position. If the cursor is at the right end of the line, it will remain there.

In addition, there are two other screen functions which are used on all terminals: backspace and linefeed. The backspace character (ASCII 8) function moves the cursor backwards, and the linefeed character (ASCII 10) function moves the cursor down one line.

The control sequences for screen functions are a single control character or an ASCII character preceded by a single lead-in character. Control sequences consisting of three or more characters are not supported.

Configuring the Screen Function Interface

Load and run the CONFIGIO program as instructed in the "CONFIGIO" section of this chapter.

Note

Before configuring the screen function interface for an external terminal, ensure that there is no accessory board installed in slot 3. If there is, turn the power off, remove the board, and use the standard Apple video output jack (see Figure 2.2 in Chapter 2). Once the configuration process is complete, you can reinstall the board and use its video output jack as before.

When the CONFIGIO selection menu appears (see page 118), press the *1* key. CONFIGIO will display the Hardware and Software Screen Function Tables as shown in the following figure:

```
+ SCREEN FUNCTION INTERFACE MENU +

    FUNCTION          SOFTWARE          HARDWARE

Clear Screen    :    ESC *        →        FF
Clr To EOS      :    ESC Y        →        VT
Clr To EOL      :    ESC T        →        GS
Lo-lite Text    :    ESC )        →        SO
Hi-lite Text    :    ESC (        →        SI
Home Cursor     :         RS      →        EM
Address Cursor  :    ESC =        →        RS
XY Coord Offst  :         32      →        32
XY Xmit Order   :         YX      →        XY
Cursor Up       :         VT      →        US
Cursor Forward  :         FF      →        FS

        1. SOROC IQ 120 IQ 140
        2. HAZELTINE 1500/1510
        3. DATAMEDIA
        4. Other
        Q. Quit

    Select - _
```

The menu above shows the default values of the Hardware and Software Screen Function Tables. Items in the SOFTWARE column are the default control sequences of the Software Screen Function Table. Items in the HARDWARE column are the ASCII codes needed by the terminal hardware to perform the stated screen function. A NUL entry in either table indicates that the function is not available.

Three of the numbered entries in the lower section of the screen are for terminals that CONFIGIO has data for. To configure the screen function interface for any of the terminals listed, type the menu number corresponding to the terminal. For terminals not listed, or for application programs requiring modifications to the screen function interface, press the *4* key. To return to the main CONFIGIO menu, press the *Q* key.

For application programs requiring changes to the screen function interface, the Software Screen Function Table is modified. External terminals will usually require modifications to the Hardware Screen Function Table.

The Software Screen Function Table must match sequences sent by the application program to perform screen functions. The Hardware Screen Function Table must have non-zero entries in all of the nine functions. We recommend setting up the Software Screen Function Table to emulate a Soroc IQ 120/IQ 140 terminal. This is a common configuration that is supported by most CP/M software.

Configuring for an External Terminal

For Soroc IQ 120 or IQ 140 terminals, no changes are needed for either screen function table. However, when you first turn on Soroc terminals, text is shown in the "highlight" mode. CP/M will reset the screen to display in a normal "lowlight" mode whenever a cold start is performed.

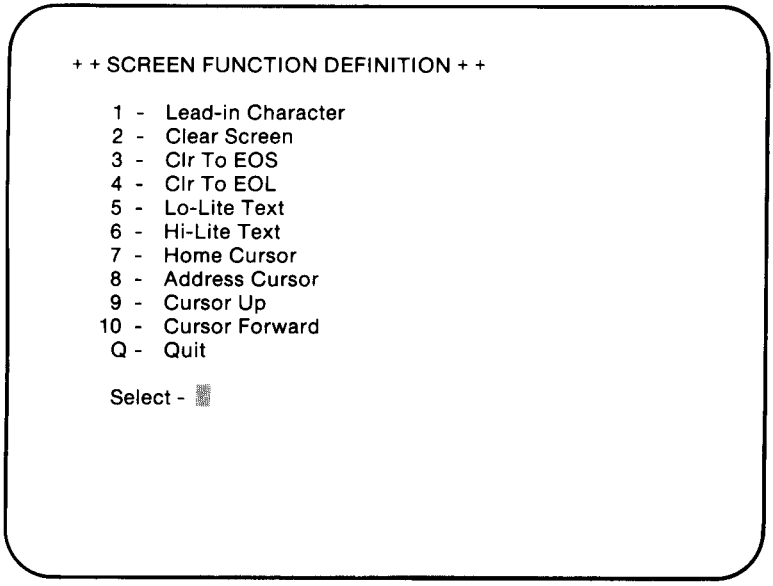
For Hazeltine 1500/1510 terminals, use the Hardware Screen Function Table only. (CP/M translates the Hazeltine cursor-addressing function with no X-Y coordinate offset.) We do not recommend using the Hazeltine screen function sequences in the software table. It is best to set up the hardware table for the Hazeltine, and the software table for another common terminal, such as the Soroc IQ 120/IQ 140.

For Datamedia terminals, set up the Hardware Screen Function Table only. (Datamedia terminal control sequences are not usually supported by CP/M software.) Set the hardware table for use with a 24x80 video board, and the software table for another common terminal type, such as the Soroc IQ 120/ IQ 140.

Note

Highlight text and lowlight text screen functions (GBASIC commands INVERSE and NORMAL) are not supported by Datamedia terminals. Thus, the table entries specified for these functions are set to an arbitrary value to ensure that these two entries will have non-zero values.

To configure the screen interface for a terminal not listed in the menu, press the *4* key when the Screen Function Interface menu appears. CONFIGIO will load and display the list of configurable screen functions shown in the figure below.



You can now change any of the values in the Terminal Screen Function Definition Table.

Note

The appropriate screen function command characters for your terminal are described in the technical manual for that terminal. To find out which codes are transmitted by a particular program (for example, a word-processing program) consult the manual for that program.

Select a number (1 through 10) to define the character sequences for any of the functions listed in Table 7.1.

Table 7.1
Screen Function Descriptions

No.	Title	Description
1	Lead-in character	Defines the lead-in character: the character (usually an ESC) that precedes the screen function command character. A particular screen function may or may not require a lead-in character.
2	Clear screen	Clears the screen and places the cursor at the top left corner of the screen.
3	Clear to EOS	Clears the screen from the cursor to the end of the screen.
4	Clear to EOL	Clears the screen from the cursor to the end of the line.
5	Lowlight text	Sets the normal video mode for displaying text.
6	Highlight text	Sets inverse or double intensity video mode, depending on which mode your terminal supports.
7	Home cursor	Puts the cursor at the top left corner of the screen, but does not clear the screen.
8	Address cursor	Tells the terminal to go to a cursor address defined by the next two characters entered.
	XY coordinate offset	Defined as part of selection 8. The XY coordinate offset is the number that is added to the X and Y coordinates when they are sent to the terminal (usually 32).
	XY transmit order	Also defined as part of selection 8. Establishes the order in which coordinates are transmitted. Must be either XY or YX (usually YX).
9	Cursor up	Moves the cursor up one line on the screen.
10	Cursor forward	Move the cursor forward on a line without deleting the character under the cursor.

To assign an escape sequence to any of these functions, type the corresponding number and press the RETURN key.

For example, press the *1* key if you wish to specify a screen function lead-in character. The program will display:

```
LEAD-IN CHAR:
```

Enter the lead-in character required. Characters can be typed in any one of the following formats:

aaa	where *aaa* is a 2- or 3-character ASCII name.
c	where *c* is any keyboard character.
CONTROL-*c*	where *c* is any character.
LC-*c*	LC- indicates that the following character is lowercase. Type this in place of a lowercase character if your keyboard has no lowercase characters.
&H *hh*	*hh* is the ASCII hexadecimal code (preceded by &H). Use this format if the character cannot be typed. (See "ASCII Character Codes," Appendix H, in the *Microsoft BASIC Interpreter Reference Manual.*)

After you have entered the lead-in character, the program will ask:

```
SOFTWARE OR HARDWARE (S/H)?
```

If the lead-in character is to be used in the Software Screen Function Table, press the *S* key. If the lead-in character is to be used in the Hardware Table, press the *H* key.

To define any of the other screen functions, press the number for that function. The program will prompt you for the command character for that particular function.

The program then returns to the Screen Function Definition menu and waits for you to select another number or *Q* (to quit the program). You can make as many changes to the tables as you wish in this way.

The process for the address cursor function differs somewhat. If you press *8*, address cursor, the process is the same as with the other selections, until you see the prompt:

```
REQUIRE LEAD-IN (Y/N)?
```

After you answer this prompt by pressing *Y* or *N*, the computer displays:

```
XY COORD OFFST :
```

Type a numeral for the number of spaces that are to be added to the X/Y coordinates before they are transmitted. Finally, the program asks:

```
XY XMIT ORDER :
```

If the X and Y coordinates are transmitted in the order Y then X, enter *YX*. If the coordinates have been transmitted X then Y, enter *XY*.

The program then asks

```
SOFTWARE OR HARDWARE (S/H)?
```

and then continues in the same manner with the other functions.

Configuring For Application Programs

Use the same procedure as that used for external terminals. Most application programs will give explicit instructions on how to configure the screen function interface. If a program requires changes to the screen function interface, but doesn't give instructions, use the following procedure:

1. Load and run the CONFIGIO program as instructed in the "CONFIGIO" section of this chapter.

2. When the CONFIGIO selection menu appears (see page 118), press the *1* key. CONFIGIO will display the Screen Function Interface Menu as shown on page 123.

3. Press the *4* key.

4. Select the desired function by pressing the appropriate key or keys.

5. When

   SOFTWARE OR HARDWARE (S/H)?

   appears, press the *S* key.

6. Type the appropriate control sequence listed by the application program documentation.

7. Save the changes that you have made in the screen function interface. (See the following section.)

Saving the Changes to the Screen Function Interface

Save the changes made in the screen function interface by first pressing the *Q* key. When the main CONFIGIO menu appears, press the *4* key. The program will display:

```
+ READ/WRITE I/O CHANGES MADE +

Read Or Write (R/W)?
```

Press the *W* key. The program will display:

```
Destination Drive (A:-D:)?
```

Press the *A* key to save the changes made in the screen interface on the system disk in drive A:. The program will then display the main CONFIGIO menu.

Using the Screen Function Interface From Within a Program

The screen functions listed in Table 7.1 make it possible to write programs that perform special screen functions. Table 7.2 shows the correspondence between the Software and the Hardware Screen Function Tables in memory. It lists the function number and the hexadecimal address of each entry. The internal format of the two 11-byte tables is identical.

Table 7.2

Screen Function Interface Addresses

Function Number	Software Table Address	Hardware Table Address	Function Description
	0F396	0F3A1	Cursor address coordinate offset. Range: 0 to 127. If the high-order bit is 0, the X and Y coordinates are expected to be transmitted Y first, X last. If the high-order bit is 1, the coordinates are sent X first, Y last.
	0F397	0F3A2	Lead-in character. This byte is zero if there is no lead-in character.
1	0F398	0F3A3	Clear screen.
2	0F399	0F3A4	Clear to end-of-page.
3	0F39A	0F3A5	Clear to end-of-line.
4	0F39B	0F3A6	Set normal (lowlight) text mode.
5	0F39C	0F3A7	Set inverse (highlight) text mode.
6	0F39D	0F3A8	Home cursor.
7	0F39E	0F3A9	Address cursor.
8	0F39F	0F3AA	Cursor up.
9	0F3A0	0F3AA	Cursor forward.

A NUL character entry in either Screen Function Interface Table will disable that function on the standard Apple screen monitor.

The standard Apple screen monitor supports all nine screen interface functions, independent of the Hardware Screen Function Table. However, if a Software Screen Function Table entry is zero, the function is disabled.

If the lead-in character of the Hardware Screen Function Table is OFF, the entire table is bypassed.

If a numbered table entry is zero, the function is not implemented.

If the entry has 1 as the high-order bit, the function requires a lead-in character.

An entry with the high-order bit set to zero indicates that the function does not require a lead-in character.

Keyboard Character Definition

Some CP/M application programs require the use of keys which are not available on the Apple keyboard. For example, the Apple IIe keyboard does not have a RUBOUT key. This can be resolved by redefining specific keys in the Keyboard Character Definition Table located at memory locations F3AC through F3B7.

Keyboard Character Definition Table

The Keyboard Character Definition Table supports up to six character redefinitions. Entries in the table consist of two bytes: the first byte is the ASCII value of the keyboard character to be redefined, and the second byte is the desired ASCII value of the character. Both bytes must have their high-order bits cleared.

If there are fewer than six entries in the Keyboard Character Definition Table, a byte with the high-order bit set is put at the end of the table.

Redefining Keyboard Characters With CONFIGIO

Load and run the CONFIGIO program as instructed in the "CONFIGIO" section in this chapter. When the first CONFIGIO selection menu appears, press the *2* key. CONFIGIO will display the Keyboard Character Definition menu as shown below:

```
+ + KEYBOARD CHARACTER DEFINITION + +

        RUB             →   CONTROL-H
        CONTROL-H       →   CONTROL-D
        CONTROL-J       →   CONTROL-X

ADD/DELETE/QUIT (A/D/Q) -
```

To redefine a character response for a key, press the *A* key. To delete an entry from the table, press the *D* key. Press the *Q* key to return to the main CONFIGIO menu.

When you press the *A* key, the CONFIGIO program displays:

```
CHAR:
```

Type the character or character sequence to be defined. The table entry can be typed in one of the following formats:

aaa	where *aaa* is a 2- or 3-character ASCII name.
c	where *c* is any character.
CONTROL-*c*	where *c* is any keyboard character.
LC-*c*	LC- indicates that the following character (*c*) is lowercase. Type this in place of a lowercase character if your keyboard has no lowercase characters.
&H *hh*	*hh* is the ASCII hexadecimal code (preceded by &H). Use this format if the character cannot be typed. See "ASCII Character Codes," Appendix H, in the *Microsoft BASIC Interpreter Reference Manual.*

Save the changes made to the Keyboard Character Definition Table by pressing the *Q* key. When the main CONFIGIO menu appears, press the *4* key. The program will display:

```
+ READ/WRITE I/O CHANGES MADE +
READ OR WRITE (R/W)?
```

Press the *W* key. The program will display:

```
DESTINATION DRIVE (A:-D:)?
```

Press the *A* key to save the changes made in the screen function interface on the system disk in drive A:. The program will then display the main CONFIGIO menu.

Example

CONTROL-C can be redefined as a NUL character (ASCII code 00) to prevent the user from exiting a BASIC program. This is accomplished by running the CONFIGIO program and selecting (2) Redefine Keyboard Characters from the main CONFIGIO menu.

When the Keyboard Character Definition menu appears, press the *A* key. When the CHAR: prompt appears, type:

```
CONTROL-C
```

and press the RETURN key. If the character is acceptable, the program prompts you to enter the new definition of the character with an arrow as shown:

```
CONTROL-C →
```

Now type

```
NUL
```

and press the RETURN key. If your entry is not acceptable, the computer will erase what you have just entered and wait for an acceptable character entry.

If the entry is acceptable, the Keyboard Character Definition menu is displayed again with the new definitions added to the menu.

Note

If you have followed the example, you will find that you cannot exit the CONFIGIO program with CONTROL-C.

To delete the entry just made, type *D*. CONFIGIO will display the CHAR: prompt again. Now type

CONTROL-C

and press the RETURN key. The list is displayed again with the CONTROL-C → NUL entry deleted.

Type *Q* to return to the main menu.

Notes on Keyboard Character Definition

We recommend that you delete entries to the Keyboard Character Definition Table if they do not apply to your keyboard. For example, if your keyboard has a RUBOUT key, you should delete the DEL entry.

Redefining CONTROL-C as a NUL character to prevent exiting BASIC programs with CONTROL-C is useful, but it can cause problems at CP/M command level. CONTROL-C is used by CP/M for a warm start.

Certain terminals and 80-column display boards perform their own character redefinitions. For example, the Videx™ Videoterm™ display board uses CONTROL-A to switch between uppercase and lowercase input mode. Since CONTROL-A is also used in BASIC to enter edit mode, we recommend redefining another character as CONTROL-A (such as CONTROL-W).

Nonstandard I/O Devices and User Software

The SoftCard IIe version of CP/M provides a means of using nonstandard I/O devices with CP/M or adding special I/O software through the use of "patch" areas and a programmable vector table. Instructions for using the patch areas and vector table are given in the *Premium SoftCard IIe System Programmer's Manual.*

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
Glossary

Access

An operation to obtain data from or place data into a storage device, register, or memory.

Accessory board

A printed circuit board installed in the accessory slots of the Apple IIe computer. It usually performs as an interface to I/O devices.

Active drive

The disk drive that all disk file operations are performed from or to if no other drive is specified. Also called the currently logged drive.

Address

A number representing a location where information is stored or where an I/O device is located.

Alphanumeric

Characters which include the letters of the alphabet, numerals, and other symbols used for punctuation and mathematical operations.

Ambiguous filename

A filename containing wild card characters in the filename or in the filename extension. An ambiguous filename is used to refer to more than one file at a time.

ANSI

American National Standards Institute. An organization devoted to establishing industry standards for computing and data processing.

Argument

A user entry in the command line of a command or program statement. Also called option, user entry, or parameter.

Assembler

A program that translates symbolic assembly language into binary machine language for execution by the computer.

Backup disk

A disk that contains information copied from another disk. It is used in case the original disk is unintentionally altered or destroyed.

BDOS

Basic Disk Operating System. The CP/M system module that handles disk operations.

BIOS

Basic Input/Output System. The CP/M system module that handles communication with the computer's I/O system.

Block

A basic unit of disk space allocation used by CP/M. Each disk drive has a fixed block size defined in its disk parameter block in the BIOS. A block can consist of 1K, 2K, 4K, 8K, or 16K consecutive bytes. Blocks are numbered relative to zero.

Boot

The process of loading an operating system into memory. A boot (bootstrap loader) program is a small program that automatically executes when the power is applied to the computer. The boot program loads the rest of the operating system into memory.

Built-in commands

Commands that reside in the CCP module. They can be used at any time from CP/M command level.

Call

See system call.

Calling program

A program or software module (such as the CCP) running in the TPA that executes a system call.

Card

See printed circuit board.

CCP

Console Command Processor. The CP/M software module that handles operator communication.

Character position

A location on the screen where one character can be displayed.

Character set

All of the characters that can be displayed and entered from a terminal.

Command file (.COM file)

An executable program in machine language.

Command line

A command to the computer that consists of the command word and the arguments or parameters that modify the execution of the command.

CON:

Mnemonic for the logical console device.

Console

See terminal.

Control character

A character used with another character to perform a special operation. See "Line Editing Commands" in Chapter 5 for a list of control characters used with CP/M.

CP/M command level

A mode of operation where the CCP module controls the other CP/M system module and hence the computer. The command level mode of operation allows direct commands by the operator to the operating system. Command level is indicated by the CP/M A> prompt.

Data file

A file containing information that will be processed by a program.

Debug

The process of detecting, locating, and removing errors in a computer program. Programs such as DDT help perform this task.

Delimiter

A special character, such as a comma, that separates different items in a command line.

Destination disk/file

The disk or file that information is to be copied to by the COPY, PIP, or MFT program.

DOS

The mnemonic name for disk operating system.

Editor

A utility program that creates and modifies text files. It can also be used to create document files or code for programs.

Extent

A CP/M measurement unit (usually 16K consecutive bytes) for storing data in a file.

External terminal

Refers to a terminal connected to an interface board in accessory slot 3 of the Apple IIe System. The external terminal replaces the Apple keyboard and screen monitor as the primary I/O device for operator input.

FDOS

An arbitrary area of memory consisting of the BDOS and BIOS software modules.

File specification

Also called filespec. A series of bytes that indicate the name, type, and location of a disk file.

Hex file

A printable form of a command (machine-language) file.

Instruction set

The list of all instructions which a given microprocessor will understand and execute.

I/O

Input/output. The transfer of data into and out from a computer and its peripheral devices.

I/O Bus

The communication circuits between the Apple CPU and the other components of the computer system.

I/O devices

The hardware devices of a computer system used to enter data into the computer, such as a keyboard; or to accept data from the computer, such as a printer.

I/O port

A register or set of registers used by the CPU for input or output of data to and from the I/O system.

I/O system

I/O devices such as printers, terminals, modems, etc., and the necessary interface circuits that permit communication with a computer.

Lead-in character

A character used by the computer to denote the beginning of a special function or routine.

Line editing

In CP/M, the act of editing the current command line.

List device

The I/O device (usually a printer) on which data can be listed or printed. LST: is the name of the logical list device.

Loader

A utility program that loads a machine-executable program into memory.

Logical device

The software representation of the actual physical I/O devices that the computer can communicate with.

LST:

The logical list device name.

Master disk

The disk that comes with the SoftCard package containing the CP/M operating system and all the software that is part of the SoftCard package.

Mode

A certain way of performing tasks. For example, a computer receives data from an I/O unit in either synchronous or asynchronous mode. In asynchronous mode, data is sent serially with no synchronization between the I/O unit and the computer. In synchronous mode, data is sent in synchronization with the computer's clock frequency.

Module

A set of routines and subroutines organized into a logical unit and designed to work with other software modules. In CP/M, there are three software modules: the CCP, BIOS, and BDOS.

MP/M

Multi-Programming Monitor control program. The multi-user version of CP/M.

Object code

Executable binary code (the output code of an assembler).

Object program

A source program that has been translated into object code that can be executed or "run" without any additional preparation.

Option

See argument.

Page

256 consecutive bytes in memory.

Parameter area

The memory area between addresses 000 and 0100. Used to hold important system parameters.

Patch

A short section of program code that replaces a section of another program to correct errors, make changes, or supply additional data.

Peripheral devices

See I/O devices.

Physical device

A vector location in the BIOS module that points to a specific assembly language routine for I/O communication.

Port

On a terminal or computer, the physical connection facilities (i.e., sockets, connectors and cables) to an I/O device.

Printed circuit board

Interface circuits contained on a board that plugs into the Apple IIe accessory slots for interfacing to an I/O device, additional memory, or a coprocessor.

Program-dependent

Input and output devices whose functions can be defined by a computer program.

Program level

When the operation of the computer is controlled by a program (such as GBASIC) running in memory. Any commands given by the operator are processed by the program instead of by the operating system's command module (in CP/M, the CCP). For example, when GBASIC is running, all commands are processed by the BASIC Interpreter. The resulting actions are passed to the BIOS and BDOS modules by GBASIC. The program level mode of operation is usually indicated by the program's prompt: in the case of GBASIC, the word "Ok."

Prompt

Instructions or symbols displayed on the screen to indicate what the operator should do next.

PUN:

The logical punch device name.

RDR:

The logical reader device name.

Read only (R/O)

An attribute that can be assigned to a disk or disk file. When assigned to a file or a disk, it does not allow changes to be made on the file or disk.

Read/write (R/W)

An attribute that can be assigned to a disk or disk file. It allows both read and write operations.

Record

A group of 128 bytes in a disk file.

Source file

The original file (usually an ASCII text file) in which a program is prepared prior to processing by the computer.

Spooling

The process of accumulating printer output in a file while the printer is busy. The file is printed when the printer becomes free.

System call

A request from a program or from the CCP to an assembly language routine that performs a low-level function such as displaying a character on the screen. In CP/M, the assembly language routines are stored in the BDOS module and are identified by numbers.

System tracks

The tracks on the disk reserved for the CP/M system.

Terminal

An input/output device; a terminal usually has a keyboard and monitor for entering and displaying data.

TPA

Transient Program Area: the area of memory where user programs are loaded and run.

Track

A separate recording path on a magnetic tape or disk.

Utility program

A program that enables the user to perform certain operations, such as copying disks.

Vector

A location in memory that "points" to a subroutine or another memory address.

Index

Important Addendum to the Microsoft Premium SoftCard IIe Installation and Operation Manual

Due to a redesign of the Apple IIe chassis, the following modification to the procedure described on page 20 is required for successful installation of the SoftCard IIe circuit board.

This procedure is only applicable if your Apple IIe has EMI molding along the back panel.

Follow the first nine steps of the circuit board installation procedure beginning on page 19. Replace step 10 with the following:

10a. Remove the slot 1 inserts from the back panel.

10b. Slide the front end of the SoftCard circuit board underneath the lip of the chassis.

10c. Tilt the board slightly downward and align the SoftCard edge-connector with the AUX slot connector. Move the board to an upright position until connectors are mated together. Make sure the back of the board positions to the left of the POWER ON indicator LED.

Perform steps 11 through 17 as instructed.

Appendix D
CP/M ProFile

Part No. 000-032-013

Document No. 8823-22X-00

This appendix will show how to install and use CP/M on the Apple ProFile hard disk drive. To install CP/M on the ProFile, follow these steps:

1. Set up the ProFile and format it.

2. Allocate space on the ProFile by creating a Pascal area and hard disk volume.

3. Copy ProDOS and CP/M to the ProFile. (ProFile will not work properly without ProDOS.)

Appendix D is organized for both new and experienced ProFile users. If you are a new user, we recommend going through all the procedures listed in this section. If you are an experienced ProFile user, follow the procedures that apply to your situation. For example, if you already have formatted the ProFile and have created a Pascal area, skip the first two procedures and continue with the third.

What You Need to Install CP/M on Your ProFile

In addition to the SoftCard circuit board and the items listed in the *ProFile Owner's Manual*, you will need the following disks:

- Apple PPM Startup disk
- Apple PPM Program disk
- Apple ProDOS User's disk
- Microsoft SoftCard ProFile disk
- Microsoft SoftCard Master disk

Before you start the following procedure, make sure that your ProFile is set up and operational as described on pages 1 through 16 in the *ProFile Owner's Manual*. For systems that have only one floppy disk controller card, install the ProFile interface card in slot 5. If you have two floppy disk controller cards in slots 5 and 6, install the ProFile interface card in slot 4. You may have to move the SoftCard to a different slot to accomplish this. Table 2.1 in Chapter 2 of the *Installation and Operation Manual* lists the slots available for the SoftCard.

Note

The following procedures assume that you have only one floppy disk drive. Certain steps ask you to "swap disks" (remove one disk and insert another). If you have more than one drive, you may use the additional drives instead of swapping disks. The software will automatically search all drives until it finds the file you specified.

Formatting the ProFile

Formatting not only prepares the ProFile to receive information but also deletes all data previously stored on it. If you already formatted the ProFile and have data that you don't want to lose, skip steps 4 through 11.

1. Turn on the ProFile power switch, and wait for the light on the front of the drive to stop blinking.

2. Insert the ProDOS User's disk into drive A.

3. Turn on the video monitor and the computer. When ProDOS is loaded into memory, the screen displays the ProDOS Main menu as shown in the figure below:

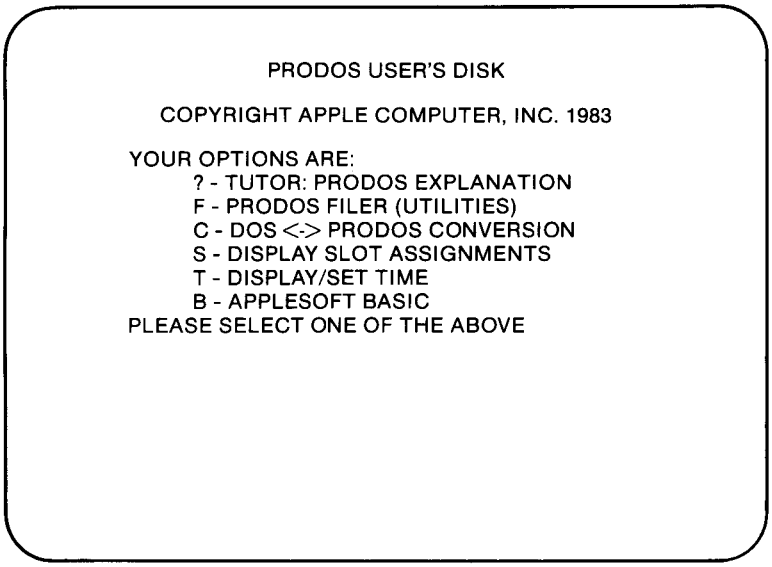


Figure D.1 ProDOS Main Menu

4. Enter *F* from the Main menu. The screen will display the ProDOS Filer menu.

5. Enter *V* from the Filer menu. The screen will display the Volume Commands menu.

6. Enter *F* (for Format a Volume), and the screen will display a message asking you to type in a slot number.

7. Enter the slot number the ProFile interface card is in (either slot 4 or 5). Another message on the screen will ask you for a new volume name.

8. To assign your ProFile a volume name, type */PROFILE* and press the RETURN key. You will see the message

   DESTROY "/PROFILE"?(Y/N)

   WARNING: YOU ARE ABOUT TO FORMAT A LARGE DISK.

9. Press the *Y* key to begin the format process. When the ProFile is formatted, you will see

   FORMAT COMPLETE

10. Press the ESC key twice to return to the ProDOS Systems Utilities Filer menu.

11. Quit the Filer menu by pressing the *Q* key and then the RETURN key. The display will show the ProDOS Main menu again.

The ProFile is now formatted and can be partitioned into different storage areas. The next task is to create a Pascal area on the ProFile.

Creating a Pascal Area on the ProFile

The next stage of transferring CP/M to your ProFile is creating a Pascal area on the ProFile and then a hard disk volume. A Pascal area is an area of the disk allocated for the Pascal operating system. Because CP/M uses a disk storage format similar to Pascal, the area allocated for CP/M must be within the Pascal area. A *hard disk volume* is a smaller storage unit within the Pascal area. Each hard disk unit is equivalent to a floppy disk, but of a variable size. Hard disk volumes are explained in the "Creating a Hard-Disk Volume" section of the *Pascal ProFile Manager Manual.*

The following steps will show you how to create a Pascal area on the ProFile, assuming that the ProFile is already formatted.

1. Insert the PPM Startup disk into floppy disk drive A (Apple drive 1).

2. Turn on the video monitor and the computer. When you see the message

   ```
   Insert boot disk
   with SYSTEM.PASCAL on it,
   then press RETURN.
   ```

   remove the startup disk from drive A and insert the PPM Program disk in its place. Press the RETURN key.

Note

If you have two or more disk drives, you can leave the startup disk in drive A and insert the program disk into drive B.

---

3. When you see the message

   Assign volumes to their default number? (Y/N)

   type *N*. You will first see the message

   Loading Pascal ProFile Manager ....

   followed by the PPM menu display as shown in Figure D.2.

```
Pascal ProFile Manager    Copyright 1983 Apple Computer, Inc.    Version 1.0

Type the first letter to select an option. Option?

V(olume Manager program

X(tended Filer program

B(ackup program

C(reate a Pascal Area on the ProFile

D(elete Pascal Area from the ProFile

Q(uit
```

Figure D.2 Pascal ProFile Manager Display

4. Press the *C* key to create a Pascal area on the ProFile.

5. When you see the message

   Create a Pascal Area ... Create Pascal area on which drive? (Enter number.)

   Enter *0* and press the RETURN key. (Drive 0 is the ProFile.) After the PPM program creates the Pascal area, the screen will show the PPM Main menu display again. Continue with the next procedure "Creating the CP/M Hard Disk Volume."

Creating the CP/M Hard Disk Volume

The following steps create a hard disk volume for CP/M.

Note

This procedure assumes that you have just completed the previous procedure. If not, insert the PPM disk into drive A and turn on the computer. You should see the PPM Main menu.

---

1. Create a hard disk volume by entering a *V* from the PPM Main menu.

   When you press the *V* key, the display shown in Figure D.3 will appear.

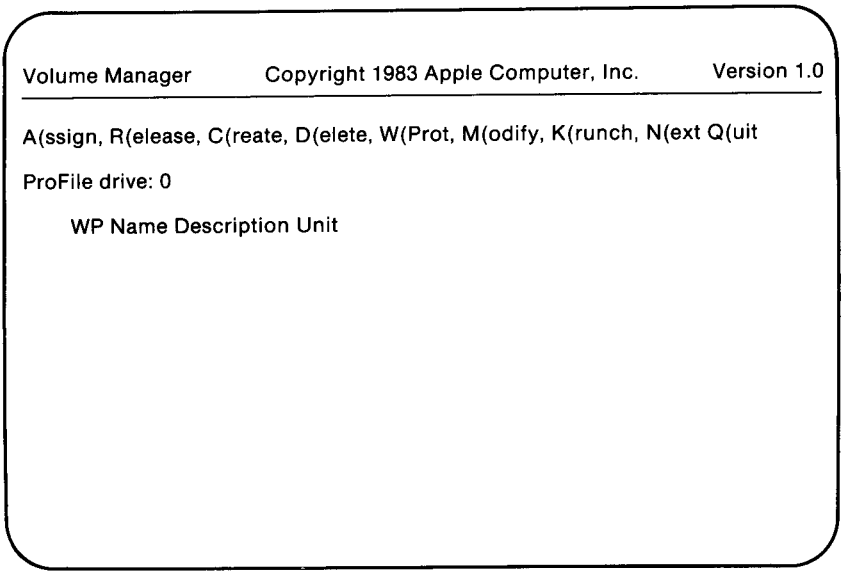


Figure D.3 Pascal ProFile Volume Manager Display

2. Select the Create command by pressing the *C* key. The screen will display the message:

   Create a volume...What is the name of this volume?

3. Type

   CP/M

   and press the RETURN key. The program will respond with the message:

   What is the description field?

4. Type

   CAT

   and press the RETURN key. The description field contains the name of the CP/M file you wish to automatically execute when the system is booted. The CP/M CAT command displays a directory of files and the amount of volume space available. CAT is given as an example, but you may use other command filenames as well.

   When you press the RETURN key, the program displays the message:

   What is the size of this volume in blocks?

5. You may enter a value equal to [(64 * *x*) + 1], where *x* is any number from 1 to 48. The minimum size then is 65 and the maximum is 9473: (64 * 48) + 1 = 9473. To create a volume of the maximum size, type

   9473

   and press the RETURN key. Note that it will take several minutes for the ProFile to create a large volume.

   Now instead of sorting through a pile of floppy disks when you want to use a program, you will go to the Volume Manager to select the volume you want to use.

6. Press the *Q* key to quit the Volume Manager program. The PPM Main menu display will appear on the screen.

Copying ProDOS onto the ProFile

The following steps copy the ProDOS operating system onto the ProFile disk and software that allows the ProFile to communicate with the SoftCard. This procedure assumes that you have just completed the previous procedure. You should have a Pascal area with a hard disk volume named CP/M on the disk.

1. Insert the ProDOS User's disk into drive A.

2. Turn the computer's power switch on. When ProDOS is loaded into memory, you should see the ProDOS Main menu.

3. Enter *F* (for Filer menu) from the Main menu.

4. When you see the Filer menu display, press the *F* key. You should see a list of file commands.

5. Enter *C* from the list of file commands. The screen will display

```
--COPY--
   PATHNAME: (                    )

     TO PATHNAME:
```

6. Type

```
/USERS.DISK/=
```

   and press the RETURN key.

7. Type

```
/PROFILE/=
```

   and press the RETURN key. The screen will display the following:

```
--COPY--
   PATHNAME: /USERS.DISK/=

     TO PATHNAME: /PROFILE/=

--INSERT DISKS AND PRESS <RET>--
```

8. Remove the ProDOS User's disk and insert the SoftCard ProFile disk into drive A. Press the RETURN key.

9. Type

```
/MICROSOFT/SOFTCARD.SYSTEM
```

and press the RETURN key. The screen will display:

```
--COPY--
   PATHNAME: /MICROSOFT/SOFTCARD.SYSTEM

     TO PATHNAME: (                )
```

10. Type

```
/PROFILE/SOFTCARD.SYSTEM
```

and press the RETURN key. The screen will display the following:

```
--COPY--
   PATHNAME: /MICROSOFT/SOFTCARD.SYSTEM

     TO PATHNAME: /PROFILE/SOFTCARD.SYSTEM

--INSERT DISKS AND PRESS <RET>--
```

11. Press the RETURN key. The screen will display the Filer menu again.

12. Exit the ProDOS Filer program by pressing the *Q* key.

13. Type

```
-/MICROSOFT/SOFTCARD.INIT
```

and press the RETURN key. This utility initializes the entire CP/M directory and performs other one-time housekeeping chores. The CP/M hard disk volume is ready for you to copy the CP/M operating system and programs onto the ProFile.

14. Type

```
-/MICROSOFT/SOFTCARD.SYSTEM
```

and press the RETURN key.

Copying CP/M Programs into the CP/M Hard Disk Volume

The following steps copy the SoftCard Master disk into the CP/M hard disk volume on the ProFile.

1. Insert the SoftCard Master disk into drive A.

2. Perform a cold start by turning the computer off and then on again.

3. When you see the CP/M A: prompt, type

   ```
   PIP C:=A:
   ```

   if the ProFile disk controller is in slot 5, or type

   ```
   PIP E:=A:
   ```

   if the ProFile disk controller is in slot 4.

4. Press the RETURN key to start the copy process. This should take about a minute to complete.

5. When you see the A> prompt again, change the active drive to the ProFile by typing either

   ```
   C:
   ```

   or

   ```
   E:
   ```

   depending on which slot the ProFile controller card is in. When you see the ProFile driver letter, use the CAT command to verify that all CP/M files from the master disk have been copied over.

This completes the last procedure for preparing the ProFile for CP/M.

How to Start CP/M from the ProFile

When you turn the computer on, it will automatically look for a ProDOS Startup file and execute ProDOS. When you see the ProDOS prompt, type

```
-SOFTCARD.SYSTEM
```

and press the RETURN key. If you used CAT in the description field, you should see a list of CP/M files from your SoftCard Master disk and a CP/M prompt (either C: or E: depending on which slot the disk controller card is in).

You may now use the CP/M operating system and CP/M application programs as described in the rest of the softcard documentation and the application program manuals.