Copy II plus

Central Point Software
INCORPORATED

WINNER
A+ READERS' CHOICE AWARDS

Over 250,000 sold!

ProDOS/DOS UTILITIES

BACKUP 5¼" & 3½" DISKS

The number one utility package for Apple II, IIGS and Laser 128 computers.

Over twenty ProDOS and DOS 3.3 utilities. Fully supports 3.5" and 5.25" disks.

Recovers accidentally deleted files. Formats disks and verifies drive speeds.

Views and prints files. Alphabetizes your disk directory. Performs disk and drive diagnostics. Uses all available memory to make fast copies without extensive disk swapping–just 1½ minutes for a 3.5" disk. Protects your software investment–backs up most protected software automatically.

Both 5¼" & 3½" diskettes inside

Central Point Software License Statement!

This software is protected by both United States copyright law and international treaty provisions. Therefore, you must treat this software just like a book, with the following single exception. Central Point Software authorizes you to make archival copies of the software for the sole purpose of backing up our software and protecting your investment from loss.

By saying, "just like a book," Central Point Software means, for example that this software may be used by any number of people and may be freely moved from one computer location to another, so long as there is no possibility of it being used at one location while it's being used at another. Just like a book that can't be read by two different people in two different places at the same time, neither can the software be used by two different people in two different places at the same time.

WARRANTY

With respect to the physical diskette and physical documentation enclosed herein, Central Point Software, Inc. warrants the same to be free of defects in materials and workmanship for a period of 60 days from the date of purchase. In the event of notification within the warranty period of defects in material or workmanship, Central Point Software will replace the defective diskette or documentation. If you need to return a product, call the Central Point Software Service Department to obtain a return authorization number. The remedy for breach of this warranty shall be limited to replacement and shall not encompass any other damages, including but not limited to loss of profit, and special, incidental, consequential, or other similar claims.

Central Point Software, Inc. specifically disclaims all other warranties, expressed or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose with respect to defects in the diskette and documentation, and the program license granted herein in particular, and without limiting operation of the program license with respect to any particular application, use or purpose. In no event shall Central Point Software be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential or other damages.

GOVERNING LAW

This statement shall be construed, interpreted, and governed by the laws of the state of Oregon.

The word Apple and the Apple logo are registered trademarks of Apple Computer, Inc.

Apple Computer, Inc. makes no warranties, either expressed or implied, regarding the enclosed computer software package, its merchantability or its fitness for any particular purpose.

ProDOS is a copyrighted program of Apple Computer, Inc. licensed to Central Point Software, Inc. to distribute for use only in combination with Copy II Plus. Apple Software shall not be copied onto another diskette (except for archive purposes) or into memory unless as part of the execution of Copy II Plus. When Copy II Plus has completed execution Apple Software shall not be used by any other program.

ProDOS Copyright 1983-1984 Apple Computer, Inc.

Table of Contents

Chapter One – Introduction

This manual describes Copy II Plus version 8, which contains *three* distinct application programs:

1. a powerful DOS/ProDOS disk utility package.
2. a sophisticated Bit Copy program for 5.25" floppy disk drives.
3. a separate Bit Copy program designed for 3.5" disk drives.

The DOS/ProDOS Utilities (which we'll just call the Utilities) allow you to manipulate DOS 3.3 and ProDOS files and disks quickly and easily. The two Bit Copy programs can make backups of valuable software that has been copy-protected, as well as making fast reliable backups of unprotected disks.

There are actually two disks in the Copy II Plus package: a 5.25" disk and a 3.5" disk. The 5.25" disk contains the Utilities and the 5.25" Bit Copy program. (There was not enough room on this disk to include the 3.5" Bit Copy program.) The 3.5" disk contains all three programs.

The Utilities

With the Utilities, you can:

Copy any 16 sector unprotected 5.25 inch disk
Copy unprotected 3.5 inch disks
Copy files
Convert files between DOS and ProDOS formats
Catalog a disk
Catalog showing file lengths
Catalog showing deleted files
Delete files
Delete all information from a disk
Lock or unlock files
Rename files
Alphabetize the catalog

Copy II Plus Disk Utilities

Format a disk
Verify that the disk is readable
Verify that files are readable
Check disk drive speed
View the contents of files
See a map of what files are stored where on the disk
Undelete files, to recover files that were accidentally deleted

These options work with either DOS 3.3 or ProDOS disks. Because of the inherent differences between DOS and ProDOS, some additional options will work only with DOS disks, and some only with ProDOS disks. The options for DOS disks include:

Copy DOS onto a disk
Catalog showing any hidden control characters
Delete DOS to free up more space for files
Change the boot program on the disk

The options for ProDOS disks include:

View any or all subdirectories without having to type pathnames
Rename a volume
Create new subdirectories

Most of the options listed above are for standard DOS 3.3 or ProDOS disks only. However, the utility options COPY DISK and VERIFY DISK can be used with any 16 sector unprotected disks, including DOS 3.3, ProDOS, SOS, CP/M, and Pascal format disks.

Note for Apple /// users: Copy II Plus is designed to work only on Apple II series (or compatible) computers. However, Apple /// SOS stores files exactly like ProDOS. You can use the Copy II Plus Utilities on an Apple II to work with Apple /// SOS disks.

The Utilities can access ProDOS files stored on floppy disks, hard disks, RAMdisks, 3.5 inch disks, and any other ProDOS-compatible "intelligent" disk devices. (This is because ProDOS itself has the ability to work with all of these devices.)

When accessing DOS 3.3 files, however, the Utilities will work only with 35-track 5.25 inch disks. (DOS 3.3, from Apple Computer, Inc., originally was designed and intended only for 5.25 inch disks. Various other companies have since devised elaborate "patches" or modified versions of DOS 3.3 to allow it to work with another type of disk device. Patches for different devices have sometimes conflicted with one another. Since Copy II Plus works from ProDOS and directly handles the 5.25 inch disks itself, there is no DOS 3.3 in memory to "patch" for these other devices.)

The Bit Copy Programs

The two Bit Copy programs are very similar. They have the same options, but one is designed for backing up 5.25 inch disks, the other for 3.5 inch disks. Except where noted, any descriptions of Bit Copy apply equally well to both Bit Copy programs.

The Bit Copy programs are provided so that you can make back-up copies of commercial programs that have been "copy-protected". Parameters for copying many programs are included right on the Copy II Plus disk. All you need to do is type in the name of the program you want to back up, and Copy II Plus does the rest! Updated parameter disks are available periodically from Central Point Software. If you want, you can also enter your own parameters to copy a disk, or use the nibble editor or hi-res disk scan options to examine how a disk is formatted.

(Note: The Bit Copy programs don't support hard disks or RAM disks, because most copy-protection schemes themselves are tied directly to the particular drive circuitry and disk format, and would not work even if you could copy the information onto a

hard disk or RAMdisk. Copy II Plus accesses the drives directly for best performance.)

Hardware Requirements & Recommendations

The hardware requirements for the 3.5" Bit Copy are slightly different from those of the Utilities and the 5.25" Bit Copy programs.

Utilities & 5.25 inch Bit Copy Requirements

To use the Utilities or the 5.25 inch Bit Copy program, you need a 64K Apple II series (or compatible) computer. This can be:

- Apple II with 16K (or larger) memory card, or
- Apple II Plus with 16K (or larger) memory card, or
- Apple IIe, or
- Apple IIc, or
- Apple IIGS, or
- Laser 128, or
- Laser 128EX, or
- other Apple–compatible computer with at least 64K of memory, and designed to work with ProDOS.

For the Utilities, you need only one disk drive. However, a second disk drive can be helpful when copying disks or files.

The COPY DISK option of the Utilities can take advantage of additional memory to copy a disk with fewer passes: a 128K computer can copy a disk in fewer passes than a 64K computer. Also, if you have at least 128K of memory, then the Utilities will look for other memory to reduce the number of passes further. (See Appendix F for information about what kinds of memory Copy II Plus can use.)

For the 5.25 inch Bit Copy program, two 5.25" drives are highly recommended. The 5.25" Bit Copy program will work with just one drive, but frequent disk swapping makes this less convenient.

3.5 inch Bit Copy Requirements

To use the 3.5" Bit Copy program, you need a 128K Apple II series "enhanced" (or compatible) computer, with 80-column display capabilities. This can be:

- "newer" or enhanced Apple IIe with 128K memory (with 65C02 microprocessor), or
- Apple IIGS, or
- Laser 128, or
- Laser 128EX, or
- other Apple-compatible computer with at least 128K of memory, 65C02 or 65816 processor, and designed to work with ProDOS.

(The 3.5" Bit Copy program does not work with Apple II, Apple II Plus, or the older unenhanced Apple IIe computers.)

You also need at least one 3.5" disk drive. The Bit Copy program can work with *any 3.5" drive except for the Unidisk 3.5.* (An inherent limitation of Unidisk 3.5 drives makes them unsuitable for bit copy type programs.) The 3.5" Bit Copy program does work with:

- Apple 3.5 drive connected to the drive port of an Apple IIGS, or
- Apple 3.5 drive or CPS 3.5 drive connected to a CPS Universal Disk Controller (UDC) card, or
- CPS 3.5 drive connected to the drive port of a Laser 128EX computer.

Depending on the amount of memory in your computer, a second 3.5" drive can be very helpful. Additional memory can reduce the number of passes needed to copy a disk. (See Appendix F for more information.)

What You Need to Know

To use the Utilities, we assume that you are generally familiar with either DOS 3.3 or ProDOS, and the standard DOS operations such as CATALOG, DELETE, RENAME, etc. If you need to know more about these things, you should refer to your DOS or ProDOS User's Manual. A few of the Utilities options are a little more complicated; this manual includes explanations for those options.

Using the Bit Copy programs to copy most protected disks doesn't require any technical knowledge, if the program you want to copy is included in our list of parameter entries. If it is not, we provide a few suggestions for how to copy new programs.

If these suggestions don't work, or if you want to learn more about disk protection schemes, then you'll need to learn and understand a number of uncomfortably technical concepts. Protection schemes are an inexact and rather sneaky art, rather than a science. Most reasonable people will not be interested in learning it. We do, however, provide some reference material on disks and disk protection in the appendices. (If you're having problems backing up a disk, remember that we also have updated parameter entries available periodically.)

Hexadecimal number notation is used throughout both Bit Copy programs and occasionally in the Utilities. (Following usual computerese conventions, the hexadecimal numbers are preceded with a dollar sign, as in "$D5".) Understanding hex numbers is helpful, but not necessary. Appendix E contains a table that lets you convert between decimal and hex.

For users interested in learning more, we recommend:

"DOS Programmer's Manual" and/or "ProDOS Programmer's Manual", by Apple Computer, for information on DOS commands, with an appendix on disk file storage,

"Beneath Apple DOS" and/or "Beneath Apple ProDOS", by Quality Software, for information on file storage and track and sector formatting,

"Understanding the Apple II" or "Understanding the Apple IIe", also by Quality Software, with a chapter of in-depth information on 5.25 inch disk hardware.

About This Manual

This manual will show you how to use each option step-by-step. In nearly every case, Copy II Plus will show "reminder" prompts as to what commands or menu options are valid. We encourage you to carefully read through this manual to take advantage of all of Copy II Plus's features.

The manual is divided into five chapters and seven appendices.

This chapter, ***Chapter One***, is an introduction to Copy II Plus, and explains how to start up the Utilities and the two Bit Copy programs.

Chapter Two describes the Utilities in depth, with information on how to use each utility option.

Chapter Three explains using either of the two Bit Copy programs to make backups of protected disks.

Chapter Four describes the "Technical Tools" found in the Bit Copy programs for those users who want to do more in-depth work.

Chapter Five tells you how to change, rename, delete, or add to the many Bit Copy parameter entries found on the Copy II Plus disk.

Appendix A is a technical reference on disks and disk hardware.

Appendix B briefly explains many disk protection schemes.

Appendix C describes the methods that the Bit Copy program uses to copy a protected disk, and discusses the various parameters used.

Appendix D is a summary of the Bit Copy parameters.

Appendix E is a table of numbers from 0 to 255, with their hexadecimal and binary equivalents, and the floppy disk 4–and–4 encoded equivalent. (Appendix A explains 4–and–4 encoding.)

Appendix F discusses how and when Copy II Plus uses extra memory in your computer.

Appendix G describes the other software backup and utility products from Central Point Software.

Starting Up

Fill out and send in your registration card now. Being a registered owner entitles you to technical support, should you need it, and it lets us tell you about product updates. If we decide to enhance or update Copy II Plus in the future, all registered owners will be able to purchase the update at a reduced price.

(**Please note:** If you purchased your Copy II Plus directly from Central Point Software, you have already been automatically registered and you do not need to send in a registration card.)

When you boot the Copy II Plus disk (either the 5.25" or 3.5" disk), the Utilities program automatically starts up. Depending on what computer set-up you have, you might need to answer a couple of questions before the Utilities main menu appears. To use the Utilities, read on to Chapter Two.

To use either Bit Copy program, first boot the Copy II Plus disk. (Remember that the 3.5" Bit Copy program is found only on the 3.5" disk.) The Utilities will start up first. You may need to press the RETURN key once or twice to skip the questions so the Utilities main menu appears. You'll notice that COPY, the first item in the main menu, is highlighted. Press RETURN to choose it. You'll then see this sub-menu:

```
SELECT OPTION:

BIT COPY 5.25
BIT COPY 3.5
FILES
DISK
DISK W/ FORMAT
DOS
```

BIT COPY 5.25 is highlighted. To go to the 5.25" Bit Copy, press RETURN to choose this option, then press RETURN again at the next prompt. The disk will whir as the 5.25" Bit Copy is loaded.

To go to the 3.5" Bit Copy, first press the down arrow key once to highlight BIT COPY 3.5. Press RETURN to choose that option, then press RETURN again at the next prompt. The disk will whir as the 3.5" Bit Copy is loaded.

Skip to Chapter Three for instructions on using either Bit Copy program.

If you have a hard disk, you may find it convenient to copy any or all of these three programs to your hard disk. The Utilities are in the file UTIL.SYSTEM. The 5.25" Bit Copy program is the file BITCOPY.SYSTEM, and it uses the parameter files PARM.KEY and PARM.DATA. The 3.5" Bit Copy program is the file BC3.SYSTEM, which uses the parameter files PARM35.KEY and PARM35.DATA.

A few words need to be said about the Bit Copy programs and copy-protected software. Under the Federal Copyright Law, you are entitled to make backups of software for your own use, so that if a disk is damaged or accidentally erased, the information is not lost. Some software companies, in efforts to prevent illegal duplication, "copy-protect" their disks so that they cannot be copied using normal copy methods. The Bit Copy programs are designed for copying these protected disks. They are provided only to help you make backups of protected disks for your own use, not for illegal copying. (Schools and institutions wishing to copy a program for educational use on a number of computers should check with the software publisher for their educational copying policy.)

The two Copy II Plus disks are standard ProDOS disks and are not copy-protected in any way. You can make a backup of Copy II Plus using the COPY DISK option in the Utilities, or with any other standard disk copy program. We encourage you to back up Copy II Plus right away, then put the original disks in a safe place, in case anything happens to your copy.

Differences from Copy II Plus Version 7

For users who are updating from Copy II Plus Version 7, here is a brief summary of the major differences in Version 8:

The 3.5" Bit Copy is a new program which was not included in earlier versions. It behaves very much like the 5.25" Bit Copy program, except that it works with 3.5" disks instead of 5.25"

disks. The Bit Copy information in Chapters Three through Five (except where noted) applies equally well to both Bit Copy programs.

The 3.5" Bit Copy program includes **"extra memory" handling**. Depending on the amount and type of memory in the computer, it can read and write numerous tracks at a time, reducing the number of times you'll need to swap disks if you have only one 3.5" drive.

Many of the parameter entries in the 3.5" Bit Copy program for Apple IIGS software include the designation "(HD)" for Hard Disk. These special parameter entries modify the backup disk made so that the program can later be **transferred to a hard disk** (with the Utilities COPY FILES option) and used from there.

In the Utilities, the principal changes were to the **COPY DISK** option. It can **optionally format the target disk** as part of the copy operation, which is useful when copying 3.5" disks. (For 5.25" disks, it always formats as it copies.) COPY DISK also includes **"extra memory" handling**, to take advantage of most extra memory in your computer to reduce the number of disk swaps. If the entire disk image can fit into memory at one time, then Copy II Plus can make **multiple copies of the disk** without having to reread the original source disk each time! This is an improvement over the old "disk-image" copying used by earlier versions of Copy II Plus, so disk-image copying is no longer supported.

A few miscellaneous minor bug fixes were also made in the Utilities.

Updated parameter entries are included with the 5.25" Bit Copy program. The Bit Copy program itself is basically unchanged, although new routines have been added to make it more powerful in making backups of copy-protected software. It copies the disk by reading and writing one track at a time.

Appendix A of this manual now includes **technical information about 3.5" disks**, as well as for 5.25" disks.

The empty example subdirectories are no longer included on the 5.25" disk, because of the limited amount of free space on that disk. The subdirectories are included on the 3.5" disk.

Chapter Two – Utilities

When you boot Copy II Plus, the disk will whir for several seconds as the Copy II Plus Utilities are loaded into the computer. What you see next depends on what kind of computer and options you have.

Setting the Date

If you do not have a Thunderclock or other compatible clock card installed in your Apple, Copy II Plus begins by asking you to enter today's date:

```
            COPY ][ PLUS 8.n
(C) 1982-7 CENTRAL POINT SOFTWARE, INC.
---------------------------------------

ENTER DATE OR PRESS [ESC] TO SKIP

DD-MMM-YY
```

When you're working with ProDOS disks, Copy II Plus can "stamp" the date onto any files you're copying. If you're using an Apple IIGS or if you have a Thunderclock or compatible clock card, Copy II Plus determines the date automatically. If not, you have the option of entering the date yourself. Type in the 2–digit day of the month, the 3–letter abbreviation for the month, and the last 2 digits of the year. (If you make a mistake, you can press the left arrow key to back up and type it again.)

If you instead press [ESC] to skip the date, Copy II Plus will simply stamp "<NO DATE>" as the date on any ProDOS files you copy.

80-Columns

Next, if you have an Apple //e computer with an 80-column card installed in the auxiliary slot, Copy II Plus asks:

```
DO YOU WANT 80-COLUMN CATALOG
DISPLAYS (Y/N) ? Y
```

If you're using a TV or color monitor, you might not want to use the 80-column display. In that case, press [N] for No. Otherwise, press [Y] or [RETURN] (or [ESC]) for Yes. When you ask for a CATALOG, Copy II Plus will then show you an 80-column catalog display.

The Main Menu

After you answer (or skip) any questions that appeared, the main Utilities menu will appear next:

```
           COPY ][ PLUS  8.n
(C) 1982-7 CENTRAL POINT SOFTWARE, INC.
-----------------------------------------

COPY                        USE ARROW KEYS
CATALOG DISK                & [RETURN] TO
DELETE                      SELECT FUNCTION
LOCK/UNLOCK FILES
RENAME
ALPHABETIZE CATALOG
FORMAT DISK
VERIFY
VIEW FILES
DISK MAP
CHANGE BOOT PROGRAM
UNDELETE FILES
CREATE SUBDIRECTORY
SET PRINTER SLOT
QUIT

                                      PRINTER
15-DEC-87                              OFF
```

Along the left side of the screen are the 15 main options. With some of these options are sub–menus to select specific functions.

Throughout Copy II Plus, you can press the Escape key ([ESC]) to back safely out of the sub–menu or current option.

One of the menu items is always highlighted using inverse (black–on–white) letters. If you want to choose that option, just press [RETURN]. If you want to choose another option, pressing the arrow keys will move the highlight inverse field to that option. Try pressing the arrow keys a few times. The left arrow (and up arrow, if your computer has one) moves the inverse field up, and the right (and down) arrow moves it down. Once the option you want is highlighted, then press [RETURN] to run it. Note the prompt in the upper–right:

```
USE ARROW KEYS
& [RETURN] TO
SELECT FUNCTION
```

At the bottom of the screen, you see either today's date, or the phrase "<NO DATE>" (if you pressed [ESC] to skip the date question). To the right you see the phrase "PRINTER OFF". By using the PRINTER SLOT option described below, you can select to use the printer with the Utilities.

One of the most convenient aspects of the Utilities is the easy way you can select which disks or files you want to work with. This is explained in the following sections under CATALOG and COPY FILES, but applies to all of the Utilities options.

PRINTER SLOT

Copy II Plus will let you print the displays from CATALOG, VIEW FILES, and DISK MAP if desired. If you intend to use the printer with the Utilities, you first need to tell Copy II Plus what slot your printer is connected to. To do this, use the arrow keys and [RETURN] to choose the PRINTER SLOT option from the main menu.

The word "OFF" underneath the PRINTER label will change to "SLOT 0" and the "0" will flash. Type the slot number that the printer interface card is in. If you decide you don't want to use the printer, press [RETURN] or type "0". The zero is used to designate "no printer", since printer cards cannot be used in slot 0. After selecting the printer slot, the main menu will become active again. (If a printer was not selected, the slot number will change back to "OFF").

(Note for Apple IIc, Apple IIGS, and Laser 128 users: If you have a printer connected to the built-in printer port on your computer, then you should select SLOT 1 to use the printer from the Utilities.)

This PRINTER SLOT option affects printing only within the Utilities program. You don't need to set this option in order to print within the Bit Copy programs.

CATALOG

To get a catalog of the disk, select the CATALOG option. A sub–menu will appear on the right of the screen. The options are:

Normal
With file lengths
With deleted files
With hidden characters

Once again, use the arrow keys and [RETURN] to choose the option you want.

Selecting Disks

Copy II Plus now needs to know which drive you want to work with. You see a display similar to the following:

```
SELECT DEVICE:

SLOT 3  DRIVE 2: /RAM DISCONNECTED
SLOT 6  DRIVE 1
SLOT 6  DRIVE 2


PRESS [?] TO DISPLAY VOLUME NAMES
```

The list of slots and drives you will see includes all floppy drives and compatible ProDOS disk devices that are plugged into your computer. Computers such as the Apple IIc, Apple IIGS, and Laser 128 have disk "ports" in addition to, or instead of, real

slots. Here are the usual "slot" assignments made by different computers for drives connected to a drive port:

- The Apple IIc and Laser 128 built-in drive are slot 6 drive 1. The drive connector is slot 6 drive 2.
- The Laser 128EX drive connector is slot 6 drive 2 if a 5.25" drive is connected, or slot 7 drive 1 if a 3.5" drive is connected.
- The Apple IIGS drive connector is slot 6 drives 1 & 2 for 5.25" drives, slot 5 drive 1 for the first 3.5" drive. If a RAMdisk is set up with the Control Panel, the RAMdisk is slot 5 drive 2 and a second 3.5" drive is slot 2 drive 1. Without a RAMdisk, a second 3.5" drive is slot 5 drive 2.

The drive selection menu works very much like the Utilities main menu. To choose a drive, all you need to do is use the arrow keys so that the desired drive is highlighted, then press [RETURN].

In addition, from this menu you can press [?] (you don't have to hold down the SHIFT key) to see what disks are in each drive. Suppose the Copy II Plus disk is in drive 1 and a DOS 3.3 disk is in drive 2. If you press [?], the drives will whir and you'll see:

```
SLOT 3  DRIVE 2: /RAM DISCONNECTED
SLOT 6  DRIVE 1: /COPYIIPLUS
SLOT 6  DRIVE 2: DOS 3.3
```

Any ProDOS disks will show a slash "/" and its ProDOS volume name. Any DOS disks will show "DOS 3.3", since DOS disks don't have volume names.

If the disk you wanted is not in any drive, you can change disks, then press [?] again to make sure the correct disk is now in the drive. When the disk you want to work with is in an appropriate drive, use the arrow keys to highlight that drive, then press [RETURN]. (You can also press [ESC] to back out if you decide you don't want to do this option.)

The "/RAM DISCONNECTED" designation in the display refers to the special small RAMdisk that ProDOS installs in any Apple II series (or compatible) computer that has 128K of memory. "/RAM" is an area of memory that ProDOS sets aside to act like a very fast disk drive, pretending to be slot 3 drive 2. If /RAM didn't contain any files (it won't if you just booted the Copy II Plus disk), then Copy II Plus disconnects it so that it can use the extra memory for its own uses. You can reconnect /RAM with the FORMAT option, described later.

Note: Copy II Plus will disconnect only the small built-in RAMdisk. If you have an "auxiliary-memory" based memory board plugged into your Apple IIe or IIc, and you've run the RAMdisk driver program supplied with your memory board, Copy II Plus won't disturb the large RAMdisk it creates. See the section on "Large Memory Boards" at the end of this chapter, and your memory board user's manual, for more information.

Selecting Subdirectories (ProDOS only)

What happens next depends on what kind of disk you've selected. If you're working with a ProDOS disk that contains one or more "subdirectories", then an additional display appears so that you can select which subdirectory you want. This is called the subdirectory "tree" display. The Copy II Plus 3.5" disk itself contains a few subdirectories. These subdirectories are empty (do not contain any files) and are included only so that you can see the tree display. If you have a 3.5" drive, then as an example, make sure the Copy II Plus 3.5" disk is in the drive, and select the CATALOG (NORMAL) option and the appropriate drive.

The disk will whir for a few moments, then the screen will display:

```
CATALOG                                                SLOT 6  DRIVE 1
/COPYIIPLUS

    COPYIIPLUS-------->SUB1
                       |
                       |->SUB2------------->SUBSUB1
                                            |
                                            |->SUBSUB2
```

This diagram shows that the main, or "root", directory (which goes by the volume name "COPYIIPLUS") has two subdirectories, named SUB1 and SUB2. In addition, the subdirectory SUB2 contains two subdirectories of its own, called SUBSUB1 and SUBSUB2. (This is called a "tree" diagram because the subdirectories extend from the "root" directory like branches of a tree.)

By using the arrow keys, you can select any subdirectory. You can move up, down, right, or left, following the "branches" between directories. (If you have an Apple II or II Plus, press [A] to move up and [Z] to move down.) If the diagram is too large to fit on the screen, it will scroll automatically as you move around on the tree.

When the directory you want is selected, press [RETURN] to actually see the files in that directory.

NORMAL CATALOG

The "normal" catalog is similar to the standard DOS "CATALOG" or ProDOS "CAT" command. Copy II Plus checks the disk in the drive to see whether it is a DOS or a ProDOS disk. If it's a DOS disk, the disk volume number is shown, then for each file, the optional "locked" asterisk, the filetype letter, the file length (in sectors), and finally the filename are shown, one line for each file. If it's a ProDOS disk, Copy II Plus lists the volume name, then for each file, the optional "locked" asterisk,

the filename, the 3-letter filetype abbreviation, the file length (in blocks), and the date the file was last modified.

If the printer is on (selected with the main menu PRINTER SLOT option), you'll first be asked whether or not you want a printout of the catalog. Answer Y (yes) or N (no).

The catalog pauses after every 20 files. You can continue by pressing any key (except [ESC], which will stop the catalog and return you to the main menu). If the catalog is being sent to the printer, it will not pause.

WITH FILE LENGTHS

The catalog "with file lengths" shows all the same information as the normal catalog. For all Basic files, it also shows the actual length of the program in bytes. For binary files, it shows both the starting memory address of the file and its length. Here is an example DOS 3.3 catalog listing for a couple of files:

```
*A 006 HELLO
       L1137   (L$0471)
*B 003 CHAIN
       A2056, L456   (A$0808, L$01C8)
```

This shows that the Basic file HELLO is 1137 bytes long ($471 in hex), and the binary file CHAIN has a starting address of 2056 and a length of 456 (with corresponding hex numbers in parentheses.)

For ProDOS disks, the CATALOG WITH FILE LENGTHS option is similar to the ProDOS 80-column CATALOG command, adding the "created" date, the length of the file in bytes, and any "subtype" when appropriate.

WITH DELETED FILES

The catalog "with deleted files" includes the files on the disk which have been marked as deleted, but have not yet been overwritten by a new file entry. Any deleted files are marked in this display with the letter "D" to the left of the entry. (Note that in some cases, deleted files can safely be recovered and made active again using the UNDELETE FILES option, explained later.)

WITH HIDDEN CHARACTERS

DOS 3.3 allows users to include hidden "control" characters in a filename. The catalog "with hidden characters" option allows you to see any imbedded control characters, which are normally not printed by Copy II Plus. The control characters show up as inverse characters. If the printer is on, control characters are translated to lower-case.

Since ProDOS does not normally allow hidden control characters in filenames, this option usually just prints a normal catalog with ProDOS disks.

COPY

The main menu COPY option gives you six separate choices:

Bit Copy 5.25
Bit Copy 3.5
Copy Files
Copy Disk
Copy Disk w/ Format
Copy DOS

If you want to go to one of the Bit Copy programs, choose the appropriate BIT COPY option. A prompt will appear. Insert the

Copy II Plus disk in the appropriate drive and press [RETURN]. The Bit Copy program will be loaded from the disk.

For the other four choices, COPY FILES, COPY DISK, COPY DISK W/ FORMAT, and COPY DOS, you need to select two drives:

1. The "SOURCE" drive, which will contain the original disk you want to copy from.

2. The "TARGET" drive, which will contain the disk you're copying to.

Copy II Plus will first ask you to "ENTER SOURCE DEVICE" and show you the same menu described earlier for selecting one of the drives. After you've selected the source drive, it then asks you to "ENTER TARGET DEVICE". Select this drive in the same way. If you select the same drive for both source and target, Copy II Plus will prompt you when to switch disks.

COPY FILES

The copy files option allows you to copy standard DOS and ProDOS files from one disk to another quickly and easily. You can also use it to convert files between DOS and ProDOS formats. You should have the disks in the drives before making the SOURCE and TARGET selections. (If you have only one drive, you should have the source disk, the one containing the files to be copied, in the drive.) The source drive will whir for a moment. If it contains a ProDOS disk with subdirectories, then the subdirectory "tree" display will appear for you to select which subdirectory contains the files you want to copy. Then a "file display" for the source disk will appear.

Selecting Files

The file display is used in various ways throughout Copy II Plus for selecting files to be worked with. Here it is used to determine which files to copy. Note that the first file in the catalog is highlighted (displayed in inverse). By using the arrow keys, you can cause any file in the catalog to be highlighted. If you repeatedly press the arrow keys, the display will scroll.

The prompt below the display reads:

```
[RETURN]-MARK, [D]ELETE, [E]NTER
FILENAME, NUMBER-INSERT, [G]O,
[ESC]-EXIT
```

These commands allow you to select not only which files to copy, but in what order to copy them! This is a handy feature if you want files to appear in a certain order on the catalog of a disk.

Pressing [RETURN] will place a number to the left of the current (inverse) file. The first [RETURN] will place the number 1, the second a 2, etc. These numbers represent the order the files will be copied in. If you accidentally press [RETURN] to number a file you don't want to copy, you can remove the number by moving the inverse field to that file and pressing [D] for Delete Number. You can also make insertions in the list of numbers by typing a number directly, then pressing [RETURN].

In addition, you can select one or more files by pressing [E], for Enter filename. You can type a single filename, and the program will look for that name in the catalog display and mark it with the next available number. You can also enter filename "patterns".

A pattern is a filename with one or more equals signs ("=") in it. The equals sign is a special "wildcard" character that will match any number of characters in the catalog, as long as the rest of the filename matches. For example, the pattern "AB=" will match the files "AB", "ABCDE", and "ABRAHAM". The pattern "=N=" will match the files "N", "OH NO", or any filename

containing the letter N. The pattern "=" will match anything, and can be used when you want to copy every file on the disk.

In addition, patterns can specify what filetypes to match. If you want a pattern to match only certain filetypes, finish the pattern by typing a comma, followed by the filetypes used in the catalog. For DOS 3.3 disks, these are the letters:

A – Applesoft
I – Integer
B – Binary
T – Text

For example, the pattern "=XYZ,BT" will match any file whose name ends in "XYZ" and is a binary or text file. The pattern "=,A" will match any Applesoft Basic file.

For ProDOS disks, the filetypes would be the ProDOS three–letter abbreviations. For example, the pattern "=XYZ,BIN,TXT" will match any file whose name ends in "XYZ" and is a BINary or TeXT file. The pattern "=,BAS" will match any Applesoft BASic file.

After you enter the pattern and press [RETURN], the program will scan through the display, marking all matching files. If no files match, the inverse field will return to the file that was in inverse before you pressed [E].

If you're working with a ProDOS disk that contains subdirectories, the File Display also provides you with another way of selecting which subdirectory you want to work with. If you want to look "into" a subdirectory, use the arrow keys to highlight the subdirectory name. Now press the [>] (greater–than sign) key (you don't have to hold down the SHIFT key). The disk will whir, and the selected subdirectory will replace the directory you were looking at a moment before. If you want to "back out" of a subdirectory, just press [<] (less–than sign). You'll be returned to the next higher directory level. You can use the [>]

and [<] keys to look into any subdirectory on the disk, before you decide which subdirectory you want to work with. You can select files in only one subdirectory at a time.

(Note: The CATALOG option in Copy II Plus doesn't have this [>], [<] feature. If you want to browse a number of subdirectories on a disk, one good way is to choose one of the file options, such as VIEW FILES, to bring up the file display. Then you can press [>] and [<] to look at different subdirectories.)

Continuing with COPY FILES

If you're copying files from a ProDOS disk that contains subdirectories, you can also select to copy entire subdirectories! Just use the arrow keys and [RETURN] to number the subdirectory like the files you're copying. The Utilities will then automatically copy the subdirectory and every file inside the subdirectory.

When you've selected all the files that you want to copy, press [G] for Go to begin the copy. If only one drive is being used, you will be prompted to insert the proper disk.

As the files are copied the program will check to see if any of the files already reside on the target disk. If there are duplicate filenames, you will be prompted, as in this example:

```
FILE HELLO
ALREADY EXISTS. NOW WHAT?

[C]OPY ANYWAY, [N]EW NAME, [D]ON'T COPY,
[ESC]-EXIT COPY
```

(If the duplicate file is locked, the program will say "IS LOCKED" instead of "ALREADY EXISTS".)

If you select to Copy anyway, the original will be deleted, then the new file copied. If you select New name, you will be asked to

type in a new name for the file. Selecting Don't copy will simply not copy this file, and pressing [ESC] will exit out of the entire copy option.

As the files are being copied, they are shown in the file display, with the file currently being copied shown highlighted. At the bottom of the screen, you'll also see the word "TARGET:", with the name of the file being copied. This is especially useful if you're copying files from a DOS 3.3 disk onto a ProDOS disk. Since ProDOS has more restrictions concerning filenames, Copy II Plus might have to slightly change the name of the DOS file to fit ProDOS rules. This changed filename is shown after the word "TARGET:" for your information.

Created and Modified Dates (ProDOS only)

Copy II Plus follows a certain convention for the created and modified dates of any ProDOS files it copies. These are the dates that can be seen with the CATALOG option. Copy II Plus uses the *current* date as the created date when making a copy of a file, rather than using the created date of the original file. The modified date of the copied file is kept the same as the original file. This convention is the same as that used in Apple's FILER program, but different from their Apple II System Utilities program.

This approach is intentional, as it provides useful information: If the created date of a file is more recent than the modified date, this tells you that this file is a copy of another file. The created date indicates when the copy was made, and the modified date indicates when the last changes were made to the original file. If the created date is older than the modified date, this means the file has been modified on this disk since it was placed there.

Converting Files Between DOS 3.3 and ProDOS

Converting files between DOS 3.3 and ProDOS is very straightforward with COPY FILES. To convert files from DOS to ProDOS, simply choose the COPY FILES option and select the DOS disk as the source disk and the ProDOS disk as the target disk. Select the files you want to convert and press [G] for Go. Copy II Plus will automatically recognize that the disks are of different types, and convert the files as it copies. To convert ProDOS to DOS, instead select the ProDOS disk as the source and the DOS disk as the target disk.

COPY DISK

The COPY DISK option can make fast, reliable copies of any standard unprotected 16 sector disks. (DOS 3.3, ProDOS, SOS, CP/M, and Pascal disks all use a 16 sector format.) COPY DISK can also copy unprotected 3.5 inch Apple-format disks, or copy between any two ProDOS disk devices that have the same size.

(The COPY DISK option is not intended for copying information between disks of different sizes. If, for example, you could copy all the blocks from a 140K floppy disk (including the disk "bookkeeping" blocks) onto a 1Meg RAMdisk, the RAMdisk would now "think" that it was only a 140K disk, too. That would leave the rest of the RAMdisk unusable. If you want to copy information between disks of different sizes, instead choose the COPY FILES option.)

To copy a disk, simply choose the COPY DISK option, select SOURCE and TARGET drives, insert the disks, and press [RETURN]. If for some reason you wish to stop the copying, pressing [ESC] will take you back to the main menu. If you're copying using only one drive, Copy II Plus will tell you when to insert each disk.

COPY DISK can take advantage of most kinds of extra memory in your computer to reduce the number of passes needed to copy a disk. As the COPY DISK option makes the copy, it first reads a number of tracks (or a number of blocks) from the "source" disk into memory, then writes those tracks (or blocks) to the "target" disk. It repeats this process (each called a "pass") until all the tracks or blocks are copied. If you're copying from one drive to another, the number of "passes" back and forth between the two drives doesn't really matter much. However, if you're copying with only one drive, the extra-memory handling can reduce the number of times you have to swap disks. (See Appendix F for information about what kinds of memory Copy II Plus can use.)

If the entire disk's contents can be read into memory in one pass, then when the copy is completed, you're offered the option to make additional copies of the same disk. These copies are made more quickly, since the original "source" disk doesn't have to be read again. This is ideal for duplication of disks (such as for a user group's public domain software library).

Copying 5.25 inch Disks

When copying normal 5.25 inch disks, Copy II Plus uses its own optimized floppy disk capabilities for greater speed. It copies the disk track by track, and automatically formats as it copies, so disks do not have to be formatted ahead of time. It doesn't matter whether you choose COPY DISK or COPY DISK W/ FORMAT when copying 5.25 inch disks; Copy II Plus's built-in floppy routines always format-and-write simultaneously.

There are 35 tracks on a 5 1/4 inch disk, numbered in hexadecimal from $00 to $22. As it reads or writes each track, Copy II Plus displays the track number at the bottom of the screen.

Copy Disk also checks for errors as it copies. If an error occurs, a message will be displayed showing what kind of error it is

(Read error or Write error) and what track on the disk it occurred on. The program will continue copying the rest of the disk. A read error means that one or more sectors on the source disk are unreadable. (The disk media itself may or may not be damaged.) If a write error occurs, then the media on the destination disk is most likely damaged. Double-check everything, then try again.

Even if the Copy Disk routine reads a bad sector, it will still write a "good" sector to the destination disk. That is, some of the data in that sector may be inaccurate, but an I/O error will usually not occur if that sector on the destination disk is read. If a disk is getting old and begins to create I/O errors, the data should be copied to a new disk using Copy Disk.

Copying 3.5 inch Disks

When copying 3.5 inch disks, Copy II Plus handles the information block-by-block through ProDOS, rather than track-by-track, because of the inherent differences between 3.5" drives and 5.25" drives. Choose COPY DISK if you know the target disk is already formatted; otherwise choose COPY DISK W/ FORMAT to format the target disk as part of the copy process (which takes a little longer).

Continuing with COPY DISK

When the copy is complete, and if the copy took just one "pass", you'll then see:

```
COPY SAME ONTO ANOTHER DISK (Y/N) ?
```

If you want to make a second copy of the same disk, press Y (for Yes). You'll be prompted to insert a new target disk. Insert a second disk and press RETURN. A second copy will be made onto this disk. The "COPY SAME" question will then be repeated so you can make more copies if you want.

COPY DOS (DOS 3.3 disks only)

COPY DOS is similar to COPY DISK, but it copies only the first three tracks of a disk. This is where the Disk Operating System is stored on DOS 3.3 disks. You can use COPY DOS to add DOS to a disk that was formatted with the Copy II Plus FORMAT option. (See FORMAT DISK below for more information.) You can copy a new DOS onto a disk that has somehow had its DOS tracks damaged or erased. You can also convert an initialized, or "slave" disk into a "master" disk. (The difference between initialized and master disks is not important in most applications. See the Apple DOS 3.3 manual for more information.)

To copy the DOS from one disk to another, insert a disk that contains the DOS into the source drive and the disk to "receive" the DOS into the target drive, then choose the COPY DOS option. The DOS will be copied onto the target disk.

(If you choose COPY DOS and select a ProDOS disk by mistake, Copy II Plus will simply inform you that this is "NOT A PRODOS FUNCTION".)

DELETE

The main DELETE option has three sub–options:

Delete files
Delete disk
Delete DOS

DELETE FILES

This option is equivalent to the standard DOS or ProDOS "DELETE" command, except that a number of files can be deleted at one time. Choose the DELETE FILES option, then

select the appropriate drive. If the disk contains ProDOS subdirectories, the subdirectory "tree" display will appear so that you can select the subdirectory in which you want to delete files.

A File Display appears next, similar to the one used in COPY FILES. As before, you can use the [>] and [<] keys to further browse through any subdirectories on the disk before selecting files to delete. The prompt reads:

```
[RETURN] TOGGLES MARKER, [E]NTER
FILENAME, [G]O, [ESC]-EXIT
```

Pressing [RETURN] causes an arrow "->" to appear to the left of the file entry. The arrow marks the file to be deleted. Repeatedly pressing [RETURN] toggles the arrow on and off. A number of files are marked by using the arrow keys and [RETURN]. A filename or pattern can also be entered with [E]. The rules for the pattern are the same as for COPY FILES. Any file that matches the pattern will be marked to be deleted.

To carry out the deletion, press [G] for Go. All files marked will be deleted. The file display will show the filenames as the files are deleted.

Note: If you want to delete a subdirectory on a ProDOS disk, first find that subdirectory in the tree display that appears. Back up one level (to the left) in the tree display, highlight that subdirectory, and press [RETURN] to bring up the file display for that higher subdirectory. Now find the name of the original subdirectory (the one you want to delete) as an entry within this file display. Press [RETURN] to mark it. If you mark a ProDOS subdirectory to be deleted, Copy II Plus will delete the subdirectory itself and every file within the subdirectory.

DELETE DISK

The DELETE DISK option cleanly erases all the "record-keeping" information on the disk, including the names and

locations of the files, and the presence or absence of DOS or ProDOS. Deleting a disk is similar to reformatting it to start over, but takes less time. (An unformatted disk, however, must be formatted before it can be used.)

The Utilities will display the disk's volume name (for ProDOS disks) or volume number (for DOS 3.3 disks). An extra warning prompt will appear on the screen to prevent data from inadvertently being destroyed:

```
READY TO DELETE DISK (Y/N) ?
```

Make sure the disk you want to delete is in the drive! Then answer "Y" to delete the disk.

DELETE DOS (DOS 3.3 only)

As mentioned above, on DOS 3.3 disks, DOS uses the first three tracks on a disk. The DELETE DOS option "frees" two of those tracks so that files can use them. The first track (track 0) is not accessible to files, and is not freed. Deleting the DOS increases the storage capacity of a disk by 8 Kilobytes, but the disk cannot be booted, since there is no longer any DOS to boot. If you try to boot a disk that has had its DOS deleted with Copy II Plus, it will print this message on the screen:

```
THIS DISK HAS NO DOS TO BOOT.

   INSERT ANOTHER DISK AND
   PRESS A KEY TO REBOOT.
```

The DELETE DOS option does not apply to ProDOS disks. You can accomplish the same thing with a ProDOS disk by deleting the file named "PRODOS".

LOCK/UNLOCK FILES

If you wish to lock or unlock one or more files, choose this option and select the appropriate drive. If you've selected a ProDOS disk that contains subdirectories, the subdirectory "tree" display will appear for you to select which subdirectory contains the files you want to lock or unlock.

Then a file display for the disk will appear. As in a normal catalog, an asterisk to the left designates each file that is locked. A new prompt is displayed:

```
[RETURN]-TOGGLE ASTERISK, [E]NTER
FILENAME, [G]O, [ESC]-EXIT
```

Use the arrow keys to select a file, then press [RETURN] to toggle its 'locked' asterisk on or off. You can use these keys to set the desired locked status for every file on the disk.

To lock or unlock a number of files automatically, press [E]. You'll be prompted for a filename, with the same pattern capabilities as discussed above. After entering a filename, you will see:

```
[L]OCK OR [U]NLOCK?
```

Press [L] to lock all of the files that match the pattern; press [U] to unlock them.

After setting all of the desired files, press [G] for Go. The catalog will be written back to the disk, with the proper files locked and unlocked.

RENAME

The main RENAME options has two sub-options:

Rename files
Rename volume

RENAME FILES

To rename files, select this option and an appropriate drive. The usual file display will appear, with yet another prompt:

```
[RETURN]-SELECT TO RENAME, [E]NTER
FILENAME, [G]O, [ESC]-EXIT
(RENAMED FILES ARE MARKED)
```

To rename a file, move the inverse field to that file with the arrow keys, then press [RETURN]. You will be asked what to rename the file as. Enter a new name and press [RETURN]. This must be a legal DOS or ProDOS filename:

- DOS filenames must begin with a letter and cannot contain a comma.
- ProDOS filenames must begin with a letter and can contain only letters, numbers, and periods.

If you enter a bad filename, the warning message "INVALID FILENAME" will appear and you will be prompted for another filename. If you decide that you do not want to rename the file, press [ESC].

For every file that is renamed, an arrow ("->") appears to the left of the file. This simply serves as a reminder as to which files have been renamed.

The Enter filename option is available, but since files must be renamed manually, the [E] option stops at the first file that matches the pattern, leaving that file highlighted. From here you can press [RETURN] to rename the file.

To make the changes permanent, press [G] for Go. The new filenames will be written to the disk.

RENAME VOLUME (ProDOS only)

If you want to change the volume name of a ProDOS disk, choose the RENAME VOLUME option and the appropriate drive. The disk's current volume name will be shown, and you'll be asked to enter:

```
NEW VOLUME NAME:
```

Type in the new name. Remember that the volume name must be no more than 15 characters long, begin with a letter, and contain only letters, numbers, and periods. Press [RETURN] to complete the renaming, or press [ESC] if you want to exit without renaming the volume.

ALPHABETIZE CATALOG

This option alphabetizes the file entries stored on the disk so that when you do a CATALOG, the files will appear in alphabetical order.

Select this option and a drive (and a subdirectory if necessary). Copy II Plus will read the current catalog, alphabetize it in the computer's memory, and show you what the alphabetized catalog will look like. If the catalog is long, you may need to press [RETURN] a few times to see the whole catalog. Then you'll see:

```
[G]-GO, [ESC]-EXIT
```

If you want this alphabetized catalog made permanent on your disk, press [G]. If you change your mind and don't want the alphabetized catalog, press [ESC]. Copy II Plus will return you to the main menu without changing the disk.

When alphabetizing bootable ProDOS disks, take note: If a disk contains two or more files whose filenames end with

".SYSTEM", the *first* .SYSTEM file in the catalog is the one that starts up whenever the disk is booted. Alphabetizing the disk can rearrange the .SYSTEM files, causing a different program to start up the next time you boot the disk! We recommend that you don't alphabetize any original program disk, that you make a backup and alphabetize the backup.

FORMAT DISK

This option formats a disk so that files can be stored onto it. A blank disk must be formatted before it can be used. If a formatted disk already contains information, then formatting it again will completely wipe out the old information.

When you choose FORMAT, a submenu appears for you to choose whether to format the disk for DOS 3.3 or for ProDOS. Then select the drive which contains the disk you want to format.

DOS 3.3 Formatting

Normally under DOS 3.3, you can use a DOS "INIT" command to initialize a disk. Formatting a disk is not quite the same as initializing one. If you're unfamiliar with the differences between formatting and initializing, here is some information that might be helpful.

The DOS 3.3 INIT command:

1. Lays down sector boundaries, dividing the disk into individual sectors, so the disk can be written to and read from (this is the actual formatting),

2. Writes the catalog track, which is a place to record the names of the files that will go on the disk,

3. Puts a copy of DOS (Disk Operating System) onto the disk so the disk will boot,

4. Saves whatever Basic program is in memory onto the disk,

5. Sets up DOS so that the Basic program will run automatically (as the "greeting" program) whenever the disk is booted.

For DOS 3.3 disks, the Copy II Plus FORMAT DISK option:

1. Lays down sector boundaries, dividing the disk into individual sectors (like the DOS 3.3 INIT command),

2. Writes the catalog track (like the DOS 3.3 INIT command),

3. Writes a "boot sector", so that if you try to boot the disk, it will print a message saying there is no DOS on this disk to boot.

Using Copy II Plus, you can make bootable DOS 3.3 disks. You will need another disk that already contains DOS and a greeting program.

1. Format the disk with the FORMAT DISK (DOS 3.3) option.

2. Use the COPY DOS option to copy the DOS from another DOS 3.3 disk onto the new disk.

3. Copy a Basic greeting program onto the disk with the COPY FILES option.

4. If necessary, use CHANGE BOOT PROGRAM (described later) to change the name of the program DOS runs to the name of the file you saved.

ProDOS Formatting

For ProDOS disks, the Copy II Plus FORMAT DISK option:

1. Lays down sector boundaries, dividing the disk into individual sectors, so the disk can be written to and read from (this is the actual formatting),

2. Writes the volume and directory area, which is a place to record the volume name and the names of the files that will go on the disk,

3. Writes "boot blocks", so that if you later copy the appropriate files onto the disk, it will become a bootable disk.

Whenever you boot a normal ProDOS disk, it first finds the file named "PRODOS" on the disk, and loads it into the computer's memory. Then it looks for the first file that ends with ".SYSTEM" (for example, "BASIC.SYSTEM"), and loads that too. The system program might look for yet a third file. For example, BASIC.SYSTEM looks for a Basic program called "STARTUP" on the disk.

You can make a bootable ProDOS disk with Copy II Plus. You need another disk that contains the file "PRODOS" and whatever .SYSTEM file you want the disk to start up with:

1. Format the disk with the FORMAT DISK (PRODOS) option.

2. Use the COPY FILES option to copy the PRODOS file and the appropriate .SYSTEM file from another ProDOS disk onto the new disk.

3. Copy any other necessary files onto the new disk.

Formatting /RAM

As mentioned earlier, with 128K Apple II series (or compatible) computers, ProDOS installs a small "RAMdisk" into part of the computer's memory. This is a program that acts like a very fast

disk drive with a volume name of /RAM. However, it stores information into Apple memory rather than onto a floppy disk. Once you turn the computer off, any files you've saved in the RAMdisk are lost.

Copy II Plus can use this same area of memory to speed up disk copying, so it "disconnects" the RAMdisk if it doesn't contain any files. That's why you see the phrase "/RAM DISCONNECTED" in the Drive Select Display. If you want to use the RAMdisk from Copy II Plus and you don't care if disk copying takes more passes, then you can use the FORMAT DISK option to reconnect /RAM. Just choose FORMAT DISK, and select SLOT 3 DRIVE 2 for the drive. After the RAMdisk is "reformatted", you can copy files to it, catalog it, and do the things you would normally do with a ProDOS disk. (If you later want to disconnect the RAMdisk again, you can use the DELETE DISK option to do this.)

Note: If you have a large (greater than 64K) "auxiliary-memory" based memory board, you need to use the RAMdisk install program supplied with your memory board before ProDOS — and Copy II Plus — can make use of the extra memory. See the section on "Large Memory Boards" at the end of this chapter and Appendix F for more information about how Copy II Plus deals with extra memory and /RAM.

VERIFY

The Verify option is used to select one of three sub-options:

Verify disk
Verify files
Verify drive speed

VERIFY DISK

This option is used to check if any sectors or blocks on the disk are bad. If the disk being verified is a 5.25" disk, Copy II Plus verifies it track-by-track. Any other disk device is verified block-by-block.

Copy II Plus quickly reads each of the tracks (or blocks) in turn. As it reads, the current track or block number is displayed near the bottom of the screen:

```
VERIFYING TRACK $03
```

If bad sectors are found on any track, their track and sector numbers will be displayed in hexadecimal in the middle of the screen, as in this example:

```
ERROR TRACK $03
SECTOR $5 7 B
```

This message means there were errors on track $03, sectors $5, $7, and $B.

When finished, the program will show the total number of errors. If you want to exit out of the verify before it's finished, you can press [ESC] at any time.

VERIFY DISK will work with standard 16 sector (DOS 3.3, ProDOS, SOS, CP/M, and Pascal) 5.25" disks and with other ProDOS compatible disk devices. Blank (unformatted) disks will produce errors, since there are no sectors written on the disk to verify. Most copy-protected disks will also produce errors, since the formatting on these disks is often different than the standard Apple 16 sector format.

If a normal DOS or ProDOS disk you're using is giving DOS I/O errors, it can be one of three things: bad data, bad sectors, or a physically damaged disk. *Bad data* means the catalog or file information is wrong, for example, telling the DOS to look for a

file on track 200! A *bad sector* is one that simply can't be read (possibly caused by a "power glitch" or by opening the drive door or pressing Reset while the drive was writing) even though the disk is still capable of storing good data. A disk can also be *permanently damaged* from improper handling, fingerprints, heat, spilled coffee, rabid dogs, etc.

It's a good idea to verify suspect disks to see where the errors are. If VERIFY FILES shows errors, but VERIFY DISK does not, then bad data stored in the catalog is the most likely culprit. If VERIFY DISK shows errors, then you have either bad sectors or a damaged disk (or a blank or copy-protected disk). You should use COPY FILES or COPY DISK to save as much of the information as you can, then try to reformat the disk. If the formatting fails, then the disk is most likely permanently damaged.

VERIFY FILES

Verify Files checks the data and sectors used by individual files. After selecting the appropriate drive (and subdirectory if necessary), the drive will whir and a file display will appear. Here, the files to be verified can be selected with [RETURN] the same way the files to be deleted were selected in the DELETE FILES option. An arrow will appear by all selected files. The Enter filename command can also be used to select files, with the usual multi-file pattern capabilities. To begin verifying those files, press [G].

The file display will highlight each file as it is verified. If an error occurs, the track and sector number (or the block number) for the error will appear. You can press [RETURN] to continue verifying the file, [SPACE] to move to the next file, or [ESC] to return to the main menu.

VERIFY DRIVE SPEED (5.25 inch drives only)

To properly read the data on disks, a 5.25" disk drive must spin at the right speed. This speed is 5 revolutions per second, or 1 revolution every 200 milliseconds. This speed was set at the factory, but with time, the drive speed can drift. If the speed is too far from 200 milliseconds, I/O errors can occur, or data can be written that is unreadable on a normal-speed drive.

The Verify Drive Speed option allows you to periodically check the speed of your disk drives. Select the option and the appropriate drive, then insert a blank or unused disk into the drive and press [RETURN]. (Do not use a valuable disk. This option writes over part of the disk!) In a few seconds, the drive speed will be displayed. Note that for normal use, the drive speed can vary from 198 to 202 milliseconds (ms.). Small fluctuations in the speed are also normal. (Strangely enough, smaller numbers mean faster speeds!) The speed will be displayed until you press [ESC].

(Note: When using the Copy II Plus Bit Copy program, you may need to adjust the speed more accurately. This is explained in the next chapter.)

VERIFY DRIVE SPEED is not used for 3.5" drives, because these drives continually change speeds depending on which track is being accessed. Circuitry built into 3.5" drives monitors and corrects their own speeds as they operate.

Adjusting Your Drive Speed

If the speed is out of bounds, in most cases you can adjust the drive speed yourself. Here are procedures for adjusting the speed of Apple Disk II or Micro-Sci A-2 drives, Duodisks, or Apple //c built-in and external drives. (Adjusting the drive speed may void the warranty. During the first 90 days of

ownership, you might prefer to take the drive to your Apple dealer for adjustment.)

To adjust the speed on Apple Disk II drives or Micro-Sci A-2 drives:

1. Turn off your computer.

2. Remove the drive cover. There are four screws on the bottom of Apple drives or on the sides of Micro-Sci A-2 drives. After removing them, slide the cover off towards the back of the drive. You might want to unplug the drive from the controller card for more room before sliding the cover off.

3. Now reconnect the drive to the controller card, and reboot your Copy II Plus disk, choosing the VERIFY DISK SPEED option.

4. The drive speed can be adjusted by turning the speed control potentiometer. This is a small ceramic box with a tiny adjustment screw at one end. It can be found on the smaller circuit board at the back of the drive (right side of the drive, far lower corner). Turn the screw with a screwdriver or your fingernail until the drive speed is correct.

5. Re-install the cover on your disk drive.

To adjust the Duodisk:

1. Tip the Duodisk up on its side so the underside of its case is exposed. There are two small holes underneath each drive, near the drive door. The speed adjustment screws are in the holes.

2. Boot Copy II Plus and choose the VERIFY DISK SPEED option. (The drive will run fine on its side. It doesn't have to be upright.)

3. With a small-blade screwdriver, turn the screw in the appropriate hole until the drive speed is correct.

To adjust the Apple //c built-in drive:

1. Tip the //c computer itself on its side so the underside of its case is exposed. There is a small hole in the bottom of the case, near the drive door. The drive speed adjustment screw is in the hole.

2. Boot Copy II Plus and choose the VERIFY DISK SPEED option. (The computer will run fine tipped on its side.)

3. With a small-blade screwdriver, turn the screw until the drive speed is correct.

To adjust the Apple //c external drive:

1. Tip the drive on its side so the underside of its case is exposed. There is probably a silver label on the bottom of the case near the drive door. Underneath this label is a small hole, and the drive speed adjustment screw is in the hole. You'll probably want to remove the label to access the hole.

2. Boot Copy II Plus and choose the VERIFY DISK SPEED option.

3. With a small-blade screwdriver, turn the screw until the drive speed is correct.

(Note: In older Franklin computers, the processor itself runs at a slightly different speed. This affects both the optimal speed for the drives and the timing of the VERIFY DISK SPEED option itself. Most Franklin drives are preset so that the drive speed reads at about 198 ms. per revolution. If you have problems accessing or backing up commercial disks on a Franklin computer, adjusting the speed closer to 200 ms. may help.)

A more technical discussion of drive speed is included in Appendix A for interested readers.

VIEW FILES

The View Files option allows you to quickly and easily look at the data in any file. This is useful for double-checking exactly what is in a file before copying it, deleting it, etc. View Files has two sub-options, for viewing the data as values or as text. The values option shows both the hexadecimal numbers and the ASCII characters in the file. The text option prints just the characters in a more readable form. In addition, if the printer is selected, the data can be sent to the printer.

To view one or more files, choose the VIEW FILES option, then the VALUES or TEXT sub-option. Select the appropriate disk drive (and subdirectory, if necessary). A file display for the disk will appear. Use the arrow keys to highlight the file you want to view, then press [G]. If the printer is selected (with the PRINTER SLOT option from the main menu), you'll be asked whether or not you want a print-out of this file. Answer "Y" for Yes to get a print-out.

The file is displayed a page at a time. You can press [RETURN] to see another page, or [ESC] to return to the catalog display.

When using the View Values option, the file is displayed as hexadecimal bytes, 8 bytes per line, with the equivalent ASCII characters to the right. Control characters are replaced with periods. In the View Text mode, the characters are printed out in standard 40-character lines. Control characters are not printed, except for carriage returns.

In the upper right portion of the screen is a running "byte count", showing how many bytes in the file have been printed. This can be used to find the approximate locations of text strings or bytes in the file.

At the end of a DOS 3.3 file, there may be a few funny characters, including inverse "@" signs. These are extra characters beyond the end of the real end-of-file. They were not suppressed because random access text files can have end-of-file markers interspersed throughout the file, before the file has actually ended. These files can still be viewed. The View Files option stops reading when there are no more data sectors to read.

When you've finished viewing one file, the program returns to the file display. From here, you can select another file to view, or press [ESC] to go back to the main menu.

DISK MAP (5.25 inch drives only)

The Disk Map gives you an informative display showing what sectors on the disk are used by which files, and which sectors are free for use. It is designed only for 5.25 inch floppy drives.

To see the Disk Map, choose the DISK MAP option and the desired disk drive. The disk will whir and you'll see a grid-like map of all the sectors on the disk, with the track numbers ($0 to $22) across the top row and the sector numbers ($0 to $F) along the left edge. (Note: If you're looking at a ProDOS disk, the sector numbers at the left will be slightly out of order. This is intentional, as it more accurately reflects the way ProDOS groups pairs of sectors together.)

In the grid, every sector on the disk that is marked as "in use" is shown as an asterisk in a white box (an inverse asterisk). Unused sectors are marked with a dot (a period). If the disk is mostly full, large areas of the grid will be filled in with inverse asterisks. You can see whether or not any given sector is in use by following the track number down and the sector number across and noting whether or not an inverse space is there.

After looking at this display, press [RETURN] again. Now a map for the first file on the disk will be displayed. The filename is shown at the top of the screen. Below, the map now shows only those sectors used by that file.

Notice the prompt at the bottom of the screen:

```
USE ARROW KEYS TO MAP OTHER FILES
```

If you press the right-arrow key, a map of the subsequent file on the disk will appear. You can use the right and left arrow keys to see a map of any file on the disk. If you're mapping a ProDOS disk with subdirectories, any subdirectory you view will appear as letter D's rather than asterisks. Similar to the file display, you can then use [>] and [<] to map the files within subdirectories. Press [ESC] when you want to exit back to the main menu.

If you've selected a printer slot from the main menu, you can also print the Disk Map to your printer. You'll be asked "DO YOU WANT A PRINT-OUT?" Answer "Y" (yes). The first disk map will be printed. If you want to print individual file maps, then bring the map you want on the screen and press [P] for Print.

CHANGE BOOT PROGRAM (DOS 3.3 only)

When a standard initialized DOS 3.3 disk is booted, it automatically runs whatever Basic program the disk was initialized with. For example, a disk that was initialized with the command "INIT HELLO" will run the program "HELLO" whenever it is booted. Using the Change Booting Program option, you change the DOS to boot a different Basic program, or even BRUN a binary file or EXEC a textfile on boot-up!

Select the CHANGE BOOT PROGRAM option and a drive that contains the appropriate DOS 3.3 disk. A file display for the disk will appear. At the bottom of the screen, the name of the file that the disk currently boots up with will be printed. To

select a new booting program, use the arrow keys to highlight the desired file. You can also Enter a filename or a pattern. The highlight will stop at the first filename that matches the pattern.

Press [G] to save this file as the booting program. Copy II Plus will automatically check the filetype of the file, and set either the RUN, BRUN, or EXEC command for boot-up.

(Note: For ProDOS disks, the first system-type file that ends in ".SYSTEM" is always loaded when you boot the disk, so CHANGE BOOT PROGRAM doesn't apply. See FORMAT DISK above, or your ProDOS User's Manual for more information.)

UNDELETE FILES

When a DOS 3.3 file is deleted, it is not immediately erased. It is instead marked internally as a deleted file, and its sectors are marked as free to be re-used. If other data does not later overwrite part of the file, it can still be recovered and made an active file. If a file has just been accidentally deleted, and no other disk writing has occurred, the file can always be recovered, or "undeleted". That is what the UNDELETE FILES option is for.

When ProDOS deletes a file, the file is marked internally as a deleted file, but with older versions of ProDOS, oftentimes some of the file "bookkeeping" information itself is lost too. There is not enough information left to guarantee that the file can be correctly recovered. In these cases, the UNDELETE FILES option will make a "best guess" effort to recover the file.

But when Copy II Plus deletes a file from a ProDOS disk, it marks the file internally as deleted, but also keeps keeps all of the file information intact. If you accidentally delete a ProDOS file using Copy II Plus, and no other disk writing has occurred, the file can always be recovered with the UNDELETE FILES option.

To undelete one or more files, select the UNDELETE FILES option and an appropriate drive (and subdirectory if necessary). A file display will come on the screen, this time containing a list of all the deleted files still stored invisibly in the catalog. (If there are no deleted files in the catalog, the message "NO FILES" will appear.) Use the arrow keys, the Enter filename command, and [RETURN] to select the files to be undeleted. Press [G].

The file display will show the files as the program attempts to undelete them. If a deleted file has already been partly or completely overwritten with other data, Copy II Plus will not undelete it, since the data is not recoverable. If any of the files cannot be undeleted, they will then be listed with the label "LOST FILES".

After the undelete is completed, use the Copy II Plus CATALOG option to look at the undeleted files on the disk. If Copy II Plus had difficulty undeleting a file (which may sometimes occur, but only for files deleted by older versions of ProDOS, not by Copy II Plus), then the undeleted file will be marked with a question mark. Copy II Plus has no way of knowing if it recovered all of the information correctly. You should always test or try using the suspect file to see if it was recovered successfully. (If the file is good, you can later lock or unlock the file to remove the question mark.)

CREATE SUBDIRECTORY

If you want to add a new subdirectory to a ProDOS disk, then choose the CREATE SUBDIRECTORY option and an appropriate disk drive. If the disk already contains some subdirectories, then the subdirectory "tree" display will appear next. You should select whichever directory level you want to add your new subdirectory to.

Next Copy II Plus will ask you to enter the new "SUBDIRECTORY NAME:". Type in the name you want to give this subdirectory. (Following ProDOS rules, it must begin with a letter, and contain only letters, numbers, and periods.) Press [RETURN] and Copy II Plus will add this new subdirectory to your disk.

QUIT

When you want to exit Copy II Plus and run another ProDOS program or boot another disk, choose the QUIT option. To boot a new disk, insert the disk and press [CONTROL–RESET]. To run another ProDOS program, insert the disk and press [Q]. The ProDOS system itself will ask you for a new prefix and for which .SYSTEM file you want to run next.

Large Memory Boards

The Utilities program supports the use of extra memory cards as ProDOS RAMdisks. This section describes how Copy II Plus treats RAMdisks as standard ProDOS disk devices. For information about how Copy II Plus uses extra memory to reduce the number of passes when copying disks, also see Appendix F.

Memory cards that can plug into or are assigned to a *numbered* slot don't need any special setup, because the card itself contains the special ROMcode that lets ProDOS — and therefore Copy II Plus — recognize it as a RAMdisk. Copy II Plus recognizes these cards automatically. Examples of numbered–slot memory cards are:

- Apple 1Meg Memory Expansion Board,
- Applied Engineering RAMFactor board,
- Laser 128EX internal slot 5 memory expansion,
- Apple IIc (newer versions) internal slot 4 memory expansion,
- Apple IIGS Control Panel (slot 5 drive 2) RAMdisk.

Memory cards that are *auxiliary-slot* based need to be set up before Copy II Plus can work with them. This is because these cards are simply large memory boards, and mean nothing to ProDOS unless a "RAMDisk driver" program is first installed to make the memory look like a RAMdisk. Examples of auxiliary-slot based memory cards are:

- Applied Engineering RAMWorks board,
- CheckMate Technology MultiMate board,
- Applied Engineering Z-RAM board (for the Apple IIc).

The auxiliary-slot memory boards come with a setup disk or utility disk. One of the programs on this disk creates the "RAMdisk driver". To use the memory board as a RAMdisk, you first need to run this program to install the RAMdisk driver into ProDOS. (See the instructions that came with the memory board.) Once this is done, the memory now appears to ProDOS as a large RAMdisk. You can then exit and switch to another ProDOS application, such as Copy II Plus. ProDOS — and Copy II Plus — will now treat the RAMdisk as any other ProDOS-compatible disk. (To start up the Utilities from Basic, you can type "-UTIL.SYSTEM".)

These large RAMdisks appear as slot 3 drive 1 or slot 3 drive 2, depending on the particular installation program. They are not disconnected by Copy II Plus, unlike ProDOS's small slot 3 drive 2 RAMdisk. If you see "/RAM DISCONNECTED" in the drive select display, that is ProDOS's small RAMdisk, not the large one set up by your installation program.

Take note, however: If you want to keep the large RAMdisk intact, you should not press CTRL-Open Apple-RESET to "cold-boot" another disk. If you do this, the RAMdisk installation you did will be lost, and you'll have to run the memory board's setup program again! You should instead "run" or "switch to" whatever ProDOS application you want to use next.

Directory Limitations

Note: For practical programming reasons, Copy II Plus can keep track of no more than 255 files in one directory at a time. If you have a ProDOS disk that contains more than 255 files in one subdirectory, the Utilities will show you the first 255. Any files after the first 255 will be unaffected by any of the operations in Copy II Plus.

(This limitation allows an entire directory to be stored in memory, giving Copy II Plus much of its speed. Copy II Plus could have been designed to handle unlimited-size directories, but that would have slowed down its operation quite a bit. Several Apple publications have recommended a maximum of 80 to 100 files in any one ProDOS directory. Directories larger than that begin to significantly bog down disk operations for *any* ProDOS application.)

In addition, the Utilities subdirectory "tree" display can keep track of at most 254 subdirectories. If you have a disk with more than 254 subdirectories, the first 254 will appear in the tree display. You can still access the extra subdirectories by first selecting a higher level directory from the tree, then using the [>] and [<] keys from the file display to "dive" into the desired subdirectory.

Chapter Three – Bit Copy

The two Copy II Plus Bit Copy programs are designed to allow you to make backups of software which, due to copy protection schemes, does not copy using standard disk duplication programs. The Bit Copy programs are easy to use, yet are capable of being adjusted to handle nearly every type of protection scheme currently in use.

The 3.5" and 5.25" Bit Copy programs are very similar. Any differences between the two are clearly labeled in the sections that follow.

Starting Bit Copy

Follow the instructions in Chapter One for starting whichever Bit Copy program you want to use (5.25 or 3.5). The disk will whir as the desired Bit Copy program is loaded.

The programs are designed to copy using only one disk controller card (one slot) at a time:

5.25" Bit Copy: Not all brands of 5.25" disk controller cards correctly identify themselves to the computer as disk controllers, so Bit Copy can't be sure of which slot to use. It will ask "SLOT NUMBER –". Type the slot number of your 5.25" disk controller.

3.5" Bit Copy: When Bit Copy 3.5 first starts up, it checks which slots in your computer are for 3.5" drives. If it finds one slot, it knows to use that slot, and the Bit Copy menu (see below) appears right away. If it finds more than one slot that can access 3.5" drives, you need to tell Bit Copy which slot it should use. It will ask "SLOT NUMBER –". Type the slot number for which 3.5" controller you want to use.

The main Bit Copy menu will appear next:

```
   COPY ][ PLUS BIT COPY PROGRAM 8.n
(C) 1982-7 CENTRAL POINT SOFTWARE, INC.
----------------------------------------

AUTO COPY
PARTIAL AUTO COPY
MANUAL BIT COPY
MANUAL SECTOR COPY
NIBBLE EDITOR
SECTOR EDITOR
HI-RES DISK SCAN
CREATE NEW PARM ENTRY
EDIT PARM ENTRY
LOAD PARM ENTRY
SAVE PARM ENTRY
RENAME PARM ENTRY
DELETE PARM ENTRY
QUIT


USE ARROW KEYS & [RETURN] TO
SELECT FUNCTION
```

Selecting a Bit Copy option works the same way as in the Utilities. One of the options is always highlighted using inverse (black-on-white) letters. Pressing the up and down (or left and right) arrow keys moves the inverse field to a different option. Once the option you want is highlighted, press [RETURN] to choose it.

Overview – Parameters

Copy II Plus can back up many protected disks automatically. However, with the increasingly complicated protection schemes used, no one automatic method can copy every disk. Some protected disks can't be copied correctly unless certain "parameters" are changed first. These parameters are values that Copy II Plus uses in deciding how to copy a disk. If you change

one or more of the parameters, this in effect tells Copy II Plus: "Don't copy the disk in the usual way; do it this way instead."

Versions 1 to 4 of Copy II Plus included a Backup Book, which listed many programs with the parameter changes needed to back them up. You would run the Bit Copy program and follow the instructions in the Backup Book for copying any particular disk. Entering parameter changes by hand was simple and easy, but it could become a little tedious after a while.

With Copy II Plus Versions 5 to 8, the parameter entries are stored right on the disk. All you need to do is select the name of the program you want to back up. Copy II Plus will look up the parameter changes for that program, make those changes for you, and copy the disk. If there is no parameter entry listed for a program you want to back up, we also provide a number of "try this" entries for 5.25" disks. Updated parameter entries are available on disk periodically from Central Point Software. The original "manual mode" is also included for typing in parameter changes yourself if you want.

About This Chapter

There are 13 separate options in the Bit Copy program. You need to use only the first one or two options when making back-ups of most disks. Other options are a little more involved, and a couple are quite technical in nature. Not everyone will want to explore the more complicated options.

To keep these differences readily apparent, we've divided these options among three chapters:

This chapter explains the basics of using the AUTO COPY and PARTIAL AUTO COPY options to make archival backups of your copy-protected disks. It also discusses some helpful things you should know when using either Bit Copy program. Anybody using Bit Copy will want to read this section.

Chapter Four describes the "Technical Tools" for those who want to control the bit copying more directly or want to examine or change the raw data stored on disk. The technical tools include:

```
MANUAL BIT COPY
MANUAL SECTOR COPY
NIBBLE EDITOR
SECTOR EDITOR
HI-RES DISK SCAN
```

Chapter Five describes how you can make changes or additions to the parameter entries stored on disk. This can be handy if you or someone else finds out how to back up a program that's not on the parameter entry list and you want to add it to the list. You can also print out the list of entries. In these options, the word "PARM" is an abbreviation for "parameter":

```
CREATE NEW PARM ENTRY
EDIT PARM ENTRY
LOAD PARM ENTRY
SAVE PARM ENTRY
RENAME PARM ENTRY
DELETE PARM ENTRY
PRINT PARM FILE
```

AUTO COPY

Choose AUTO COPY when you want to copy a program from the Copy II Plus parameter list. A new screen will appear:

```
AUTO COPY

NAME:
```

Notice the help lines at the bottom of the screen:

```
ENTER PARM ENTRY NAME OR
PRESS [RETURN] FOR LIST OF ENTRIES
```

If you know that the program you want to back up is included in the parameter list, type the name of the program and press [RETURN]. If you instead want to see the list of parameter entries, just press [RETURN].

If you press [RETURN] without entering a name, the disk will whir and a display of all available parameter entries will appear (similar to the "file display" from the Utilities). Note that the first entry name is highlighted. By using the arrow keys, you can cause any name in the list to be highlighted. If you repeatedly press the arrow keys, the display will scroll to show you all of the entries. Pressing [B] will display the beginning of the list; pressing [E] will display the end of the list. Use these keys to highlight the entry you want, then press [RETURN] to select it.

You can also select to see just a part of the parameter entry list. This is especially helpful when you're not quite sure of the spelling for the entry you want. When you're asked for the name, type in just the first few letters of the entry name, then press [RETURN]. Copy II Plus will show you only those entries that begin with the characters you typed. You can then use the arrow keys and [RETURN] to select from that list.

Once you've selected the entry name — either by typing it in or by selecting it from the list — the disk will whir again as the parameters to copy that program are loaded from the disk.

A new display appears now, for you to select which drives you'll be using for copying the disk. If you have two drives, you'll usually want to copy from the original disk in drive 1 to a duplicate disk in drive 2. You can change this if you like. If you have only one drive, you'll of course use drive 1 for both the original and duplicate disks. Copy II Plus will then tell you when to insert each disk.

On the screen you'll see:

```
ORIGINAL  DRIVE:     1
```

If you want the original disk in drive 1, type "1" or just press [RETURN]. If you want to use drive 2, type "2". The next question is:

```
DUPLICATE DRIVE:     2
```

Similarly, press [RETURN] to accept drive 2 for the duplicate disk if you have two drives, or type a new drive number.

After you've answered the DUPLICATE DRIVE question, a few other questions, along with the correct answers for copying this disk, will pop immediately onto the screen. The parameter entry you selected is filling in the answers for you! At the bottom of the screen you'll see:

```
      - INSERT DISKETTES -

RETURN TO BEGIN          Q TO QUIT
ESC    TO RESTART        / TO MODIFY
```

You don't need the "Q" or "/" commands here. They're explained later under MANUAL BIT COPY. If you decide you don't want to copy the disk, press [ESC] to go back to the main Bit Copy menu.

To copy the disk, now insert the original disk you're copying into the "original drive", and insert a blank or "scratch" disk (one you don't mind writing over) into the "duplicate drive". Press [RETURN] to start copying.

Write-Protecting Original Disks

Write-protecting a disk guarantees that the computer cannot write or change any information on the disk. Suppose that while using Bit Copy, you accidentally inserted the original disk at the

wrong time or into the wrong drive. Even if this happens, the information on your original disk can't be hurt if the disk is write-protected; the electronics in the disk drive will prevent any program from writing onto a write-protected disk.

5.25" Bit Copy: If you want to be extra safe, put a write-protect tab over the notch on your original disk before you copy the disk. (The 5.25" Bit Copy program doesn't *require* you to write-protect the original disk, because write-protect tabs aren't always conveniently at hand!)

3.5" Bit Copy: Before Bit Copy 3.5 will back up a disk, it will first insist that the disk be write-protected. To write-protect a 3.5" disk, slide the small tab in the corner toward the edge of the disk, uncovering the hole. If you insert an original disk (the disk to be copied) that is not write-protected, Copy II Plus will eject the disk and prompt you to slide the write-protect tab and reinsert the disk. (If for some reason you really want to leave the disk not-write-protected, just reinsert the disk. It won't check a second time.)

Copy Status

Copy II Plus uses the middle of the screen to give you detailed technical information about each track of the disk as it is read and analyzed. You can ignore most of this information if you want. (A description of this information is in Chapter Five.)

The bottom of the screen gives you status information about how the copy is proceeding. Copy II Plus goes through several stages in copying each of the tracks on the disk. It must "read" each track into memory from the original disk, then it must "analyze" this track. It then "writes" it out to the duplicate drive. Lastly, it must "verify" that the track was written correctly, then it can go on to the next track. For some disks, the copy process will include "synchronizing" to each track before reading or writing.

5.25" Bit Copy: The 5.25" Bit Copy program reads and writes one track at a time. There are 35 tracks on a 5.25" disk, numbered 0 to 34, or $00 to $22 in hexadecimal. (Some copy-protected disks use a 36th track, track number $23.) The bottom of the 5.25" Bit Copy screen shows a display like this:

```
                COPY STATUS
HEX 00000000000000001111111111111112222
TRK 0123456789ABCDEF0123456789ABCDEF0123
------------------------------------
ERR
+.5
```

Beside the words "HEX TRK", you can see the track numbers in columns, from 00 at the left to 23 at the right. Each column on the screen corresponds to a track on the disk. As the copy process continues, you'll see status letters, then error numbers, appear under each column, to the right of the word "ERR", keeping you informed of the progress of the copy. If this particular disk is being copied using "half-tracks", then error numbers will also appear on the next line down. It says "+.5" on this line, because that's where the half-track (.5-track) error numbers are shown.

3.5" Bit Copy: The 3.5" Bit Copy program can read and write a number of tracks at a time, determined by how much extra memory there is in your computer. See Appendix F for information on how Copy II Plus uses extra memory. Note: Bit Copy tends to use a lot of memory so that it can correctly record possible strange disk formats and protection schemes. For this reason, even if your computer has 1Meg of usable RAM, Bit Copy will not read all of an 800K 3.5" disk at one time.

There are 80 tracks on a 3.5" disk, numbered 0 to 79, or $00 to $4F in hexadecimal. For double-sided disks, each track is further divided into two sides, side 1 and side 2. (Most Apple-format 3.5" disks are double-sided.) The bottom of the 3.5" Bit Copy screen shows a display like this:

```
                              COPY STATUS
0                 1                 2                 3                 4
0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF
--------------------------------------------------------------------------------
```

You can see the track numbers in columns, from 00 at the left to 4F at the right. Each column on the screen corresponds to a track on the disk. As the copy process continues, you'll see status letters, then error numbers, appear on the track/status display under each column, keeping you informed of the progress of this copy. Two letters will normally appear under each column. The top letter is for side 1, the bottom letter for side 2.

Here are the status letters that appear for both 5.25" Bit Copy and 3.5" Bit Copy:

S	Synchronizing track (doesn't always appear)
R	Reading track
A	Analyzing track
W	Writing track
V	Verifying track

(In some cases, the verifying takes only a fraction of a second, so you may or may not be able to see the "V" in the status display.)

Errors and Error Numbers

In addition, as each track is finished, a track status (error) number will be left on the display. The numbers, and their meanings, are:

0 No error. Track copied correctly.
2 Read error. Cannot read the track with these parameters.
3 Track too long.
4 Duplicate disk is write-protected. Remove the tab.
5 Write verify error. (For 5.25" Bit Copy, most likely cause is duplicate drive speed too fast.)
6 Nibble count error.
7 Sector edit I/O error.

(Error number 1 is no longer used.)

With copy-protected software, remember that Copy II Plus is trying to copy a disk that was designed not to be copied! A couple of things to keep in mind:

1) Even if you get errors on one or more tracks (if the error number is not 0), the duplicate disk may still work. The error may be on a part of a disk that's ignored by the program anyway. Copy II Plus also tries its best to copy the track correctly even if there is an error.

2) If you don't get any errors, it's still possible that the duplicate disk won't work. Without the correct parameters set, Copy II Plus might "miss" a piece of "hidden" formatting that the program does need in order to boot.

The best test is always to boot the duplicate disk to see if it runs correctly!

Comments

When the AUTO COPY is finished, it will display the message "PRESS RETURN" at the bottom of the screen. AUTO COPY also has the capability to print a comment on the screen. If a comment was included in the parameter entry, then Copy II Plus will print the comment as part of the copy process. The comments are usually helpful hints in getting the backups to work. You might see comments like:

```
PUT WRITE-PROTECT TAB ON BACKUP
BEFORE USING.
```

or

```
IF BACKUP DOESN'T BOOT, TRY
RE-COPYING TRACK 1.
```

AUTO COPYing again

If you select AUTO COPY again while still in the Bit Copy program, it behaves a little differently. Suppose you're making two backups of a program called "VIDEO GAME". The first time, you can either type the name VIDEO GAME or select it from the parameter list. After the first copy is made, though, the parameters for copying VIDEO GAME are already loaded. When you select AUTO COPY a second time, you'll see:

```
              AUTO COPY

USE 'VIDEO GAME'? Y
```

Press [Y] for Yes, or just press [RETURN], to use the VIDEO GAME parameter entry again.

Whenever a parameter entry is already loaded in the computer, you'll be asked this question so that you can use the entry again without having to reload it.

If you instead want to AUTO COPY a different program from the parameter list, you'll need to reinsert your Copy II Plus disk so it can load the parameter list. Press [N] for No in response to the above question. Then you can select a new parameter entry name as you did before.

PARTIAL AUTO COPY

It's just another aspect of Murphy's Law that with a few of the protected disks, you may need to try copying the disk a couple of times before you get a copy that works. Because of the critical

disk timing (measured in millionths of a second) and other floppy factors, some disks will not copy exactly the same way every time.

If a backup doesn't work, quite often it's only one track or one group of tracks that wasn't copied correctly. The rest of the disk may be fine. In this case, all you need to do is recopy those tracks on the same duplicate disk. The parameter entries for these disks will usually include a comment telling you what tracks will need to be recopied. (See "Comments" above.)

Anytime you want to recopy just a range of tracks on a disk, select the PARTIAL AUTO COPY option from the main menu. PARTIAL AUTO COPY lets you choose what range of tracks to copy, but fills in the rest of the parameters for you, like AUTO COPY.

To choose PARTIAL AUTO COPY from the Bit Copy menu, use the arrow keys to highlight this option, then press [RETURN]. You'll be asked for the parameter entry name. Select the entry as you did in AUTO COPY. Next, answer the ORIGINAL DRIVE and DUPLICATE DRIVE questions.

The next question is not filled in for you as it was before. The prompt reads:

```
ENTER START TRACK:  0
```

Type in the number of the track you want to start copying on. You can just press [RETURN] if you want to start with track 0. The next question is:

```
ENTER END   TRACK:
```

Type the number of the last track you want copied, or just press [RETURN] to copy up to track $22 for 5.25" disks, or track $4F for 3.5" disks. If you enter the same number for both start and end tracks, then only the one track will be copied.

(Note: Some programs don't use every track on the disk, and the parameter entries for those programs won't copy the unused tracks. If the track range you enter is not found in the parameter entry at all, then nothing will be copied.)

Any additional questions are filled in for you as before. Insert your original and duplicate disks (or just the original if you have only one drive), then press [RETURN] to start copying. Copy II Plus will copy just the range of tracks you specified, setting all the parameters that apply to those tracks.

HD Parameter Entries (3.5" Bit Copy)

For the 3.5" Bit Copy program, many of the parameter entries for Apple IIGS software include the designation "(HD)", which means a backup made with that entry will be modified, as part of the copy process, to work on a hard disk. Simply back up the program using the AUTO COPY option, choosing the appropriate "(HD)" parameter. After the backup is made, go to the Utilities program. Choose COPY FILES, and copy the application file(s) *from the backup* to your hard disk. The program should then work correctly from the hard disk.

Note for Apple IIGS users who have Central Point UDC cards and 3.5" drives: There are a few Apple IIGS programs which do not work with a UDC card, because of trickeries in the copy protection used on the disk. The "(HD)" parameter entries work by removing the copy protection from the backup during the copy process. If an "(HD)" or "UDC" entry is available, use AUTO COPY to back up the problem disk. The backup should work correctly with your UDC!

Helpful Things to Know When Using Bit Copy

Many, many copy-protected programs can be backed up with Copy II Plus without any special parameter "changes" at all.

These programs are not included in the list of parameter entries, but can be backed up with the special entry called "TRY STANDARD". TRY STANDARD can be found in the alphabetical list of AUTO COPY parameter entries. Whenever you want to back up a program that is not included in the parameter entry list, first try copying it using TRY STANDARD. If that doesn't do the trick, read on...

We've received calls from some customers who "wish" that we could write the ultimate automatic-and-perfect bit copy program that would copy anything and everything without requiring parameter changes. We wish it were possible to do just that! But it's not. There are some very good but rather complicated reasons why there will never be one copy method which will back up every copy-protected disk perfectly. (A few of those reasons are touched on in the appendices.)

That's where the parameter entries come in. We have many technically oriented users who determine how to copy a program and are kind enough to share their discoveries with us. We make sure, as much as possible, that the contributions seem sensible enough, then include them in the next parameter disk. And, of course, we go out and buy those products that are in hottest demand and work on those ourselves!

See the end of this chapter for information on how to get updated parameters when they are available.

More Helpful Hints

Occasionally you may find that a parameter entry won't back up the program it's designed to copy. Why not? Keep in mind that the software publishers who copy-protect their products will sometimes—without fanfare—change the protection scheme used on the disk. When this happens a new parameter entry is needed to back up the disk. The "old" parameter entry will still back up the older versions of the program. Or perhaps the

supplied parameter entry is for a newer version of a program, and you have an "old" version. You'll sometimes find "alternate" parameter entries for a single program listed, for this very reason.

In addition, the disk copy process itself is not perfect. The Apple disk drive circuitry has a sense of humor all its own, and won't always read the same track exactly the same way every time. There are also subtle differences between disk drives. (For example, some 5.25" drives can handle "quarter-tracking" a little better than others.)

There are two things you can do which will often help:

1. Try it again! If your backup of a program doesn't work, try copying it again. If you get errors on certain tracks, try recopying just those tracks again using PARTIAL AUTO COPY. If you have two drives, also try reversing the direction of the copy (copy from drive 2 to drive 1).

2. For 5.25" disks, check your drive speeds. The speed of your duplicate drive is much more critical than the speed of the original drive. *Ideally*, your duplicate drive should be spinning at the same speed as *the drive which originally made the disk you're trying to back up,* which is a little hard to determine! As a more general rule, bit copy programs will work best if the duplicate drive is slowed down a little. If you get error 5's when backing up a disk, your duplicate drive is probably too fast for bit copying.

You can use the Copy II Plus Utilities VERIFY DRIVE SPEED option to check your drive speeds. The optimum speed for normal use is 200.0 MS. (Larger numbers, strangely enough, mean slower speeds.) If you have two drives, we suggest you set your original drive to spin right at 200.0 MS., which is perfect, and set your duplicate drive to spin at 201.0 MS., which is well within tolerance and will also help the bit copy process. (In a few cases, when noted in the parameter entries, you may need to slow your duplicate drive further to make a working backup. In

addition, with a few programs that use the "synchronized tracks" option, it IS best to have both original and duplicate drives spinning at close to the same speed.)

The 3.5" drives monitor and dynamically adjust their own speed track by track as they operate, so there is no need to adjust the drive speed of a 3.5" drive.

A number of protected programs check the write-protect notch when they start up. If your original disk is write-protected, then write-protect your backup before you begin using it.

TRY Entries

If the provided parameter entry still won't make a working backup — or if the program you want to copy is not in our parameter list — there are several special parameter entries that you can try. Each of these entries begins with the word "TRY", as in "TRY HEADER" and "TRY SYNC", and can be found in the alphabetical list of AUTO COPY parameter entries.

These TRY entries are designed to copy many protected disks. Each one uses a slightly different copy method. Select one of the TRY entries with AUTO COPY, and use that entry to try backing up your disk. Hopefully one of the TRY entries will back up your disk.

"What Else?"

If the TRY entries don't work, then a new parameter entry is needed to back up the program. Some of our users will be interested in determining parameters for themselves; many others won't. As we said earlier, figuring out copy-protection schemes can be awfully complicated; it's not for everybody. We provide some information in the appendices to help you, but there is no

definite "guide" that can be followed. The people who design copy-protection schemes try to make it as difficult as possible!

As new parameters are developed by either Central Point Software or many of our customers, we add these new parameters to the list. As mentioned earlier, we've gotten a terrific response from contributors so far. We expect to make updated lists available every few months; you, of course, can update when you choose. See the end of this chapter for information on how to update.

If none of the above methods work, you might want to drop us a quick note to let us know the name of the program and the program version number. We'll add it to our request list. Also, if you've discovered how to back up a program, let us know – we may add your contribution to our next parameter disk! Include 1) the name of the program you copied, 2) the publisher's name, 3) the version number, if any, and 4) what copy method you used.

QUIT

Choose the QUIT option from the main Bit Copy menu when you want to exit out of Bit Copy and run another program. To go back to the DOS/ProDOS Utilities, press [RETURN]. To run another ProDOS program, press [Q]. The ProDOS system itself will ask you to enter a new prefix and the name of the .SYSTEM file you want to run next. To boot another disk, insert the disk and press [CONTROL-RESET].

Updating Copy II Plus

Updates to Copy II Plus are made available at the following special prices:

Major Program Enhancements:

$15 and $3 postage and handling ($8 postage and handling if outside North America).

Updated Parameter Files:

$5 and $3 postage and handling ($8 postage and handling if outside North America).

Registered owners will be notified by mail when these updates are available so be sure to send in your registration card! If you purchased Copy II Plus directly from Central Point Software, you were registered automatically.

Chapter Four – Bit Copy "Technical Tools"

MANUAL BIT COPY

MANUAL BIT COPY is the option to use if you want to set the parameters yourself before copying a disk. Perhaps you have parameters for backing up a program written down on paper, but not yet stored as a parameter entry on disk. Or if you're familiar with the Copy II Plus parameters, you may want to experiment with changing them while copying disks. MANUAL BIT COPY lets you enter these changes.

When you select MANUAL BIT COPY from the menu, the usual Bit Copy screen will appear. You'll be asked to enter:

```
ORIGINAL  DRIVE:
DUPLICATE DRIVE:

ENTER START TRACK:
ENTER END   TRACK:
TRACK INCREMENT:

SYNCHRONIZE TRACKS?
KEEP TRACK LENGTH?
```

The 3.5" Bit Copy also includes:

```
NUMBER OF SIDES:
```

If you make a mistake when answering any of these questions, press [ESC]. You can then go through the questions again.

The first four prompts have been discussed earlier. Select which drives you want to use for the original and the duplicate disks. Then enter the start and end tracks for the range you want to copy. To copy the entire disk, just press [RETURN] twice to accept a start track of $0 and end track value of $22 ($4F for 3.5" disks).

The next question, TRACK INCREMENT, determines what kind of spacing to use. Most disks use adjacent tracks (tracks 0, 1, 2, 3, etc.). These are copied with a track increment of 1.

5.25" Bit Copy: Apple 5.25" drives can be positioned to read from any half-track or even quarter-track boundary. The only limitation is that in most cases, to work reliably the tracks of information must be spaced at least one track increment apart. For example, a protected disk could use tracks 0, 1.5, 3, 4.5, etc. This would be copied with a track increment of 1.5. You can enter half-tracks and quarter-tracks in response to the START TRACK, END TRACK, and TRACK INCREMENT questions. Half-tracks are numbers that end in ".5"; quarter-tracks end in ".25" or ".75".

3.5" Bit Copy: Apple 3.5" drives can step only whole tracks at a time. The START TRACK, END TRACK, and TRACK INCREMENT values can be any whole numbers such as 1, 2, 3, etc. The question NUMBER OF SIDES is also included. In most cases, just press [RETURN] to accept 2 as the number of sides. If you have a single-sided disk you want to copy, press 1 instead.

The next question for both Bit Copy programs is SYNCHRONIZE TRACKS? If you answer [Y] for Yes, Copy II Plus will maintain the track-to-track alignment of the data from the original disk to the duplicate. Synchronizing tracks slows down the copying somewhat, so you'll probably want to use it only when you think the disk you're copying requires it.

The last question is KEEP TRACK LENGTH? This is also known as "nibble counting", and if selected, it will cause the duplicate disk to have the same number of "nibbles" per track as the original disk. Nibble counting will help back up disks that require it, but takes longer and can otherwise make the disk slightly less reliable. Answer [Y] for Yes if you want to keep the track length.

(Note: For interested readers, more information on track spacing, synchronized tracks, and nibble counting can be found in Appendix B.)

After you've answered all of these questions, you'll see the same prompt at the bottom as before:

```
        - INSERT DISKETTES -

RETURN TO BEGIN          Q TO QUIT
ESC    TO RESTART        / TO MODIFY
```

Press [Q] if you want to quit out of the Bit Copy program altogether and boot another disk. Press [ESC] if you want to go back to the Bit Copy main menu.

You may need to change one or more parameters before copying the disk. Every parameter has both a parameter number and a value. For example, parameter number $31 determines whether or not Copy II Plus will fix "invalid" bytes on the disk. If the value of parameter $31 is 1, then Copy II Plus will fix invalid bytes; if the value of parameter $31 is 0, then it won't. Other parameters have different effects. (Each parameter is explained in Appendix C.)

To change parameters, press the [/] (slash) key. You'll see:

```
        - PARAMETER CHANGE -

    CHANGE WHAT PARAMETER:
```

Type the number of the parameter you want to change and press [RETURN]. Copy II Plus then asks:

```
            TO WHAT VALUE:
```

The current value of the parameter is displayed under the flashing cursor. To change it, type the new value and press [RETURN]. If you want to keep the current value, just press [RETURN].

After you've entered the new value, it will go back to the CHANGE WHAT PARAMETER question so that you can change another parameter. When you're finished changing the parameters you want, just press [RETURN] instead of typing a parameter number.

Now you'll be back to this menu:

```
        - INSERT DISKETTES -

  RETURN TO BEGIN        Q TO QUIT
  ESC TO RESTART         / TO MODIFY
```

Insert the disk you want to copy into the 'original drive' and insert a blank disk into the 'duplicate drive'. Press [RETURN] to begin copying.

As each track is copied, you'll see the copy status letters and error numbers appear across the bottom of the screen (described earlier under AUTO COPY). Additional technical information (see the appendices) appears in the middle window. It may look something like:

```
------------------------------------
TRACK: 00  START: 6C48  LENGTH: 1824

       FF FF FF FF FF FF FF FF
       D5 AA 96 FF FE AA AA AA
       AA FF FE DE AA EB FF FF
       FF FF FF FF FF FF D5 AA
       AD B6 DB DC F4 F3 BB BD
       CF 97 9A AE AE 96 AD AC
       9A AB 97 B2 B2 AD AB 9A

  SOURCE: 1881    OBJECT:       SYNC
------------------------------------
```

The TRACK number simply tells you which track is being copied. The START value is the address within the memory buffer that Copy II Plus found the start of the track. The LENGTH value is how many bytes long (minus any "big gap") the track data is.

Next is a block of hexadecimal bytes from the disk which Copy II Plus determined to be the track start. "Sync" bytes are shown in inverse, and the actual track start is the first byte in the second row.

On the last line, the SOURCE number is the total number of bytes on the original track, including a possible sync field before the data. A number will also appear for OBJECT, showing the number of bytes that were written to the duplicate disk. When nibble counting is used (when you answer Yes to the KEEP TRACK LENGTH question), this number will change as Copy II Plus adjusts the number of bytes being written to match the SOURCE byte count. On the right, you'll see either "HEADER" or "SYNC" for each track. This describes which method Copy II Plus used to determine the start of the track.

MANUAL SECTOR COPY

The MANUAL SECTOR COPY option provides an alternate way of copying some protected disks. Rather than reading an entire track at a time, MANUAL SECTOR COPY reads each sector from the track. It then formats and writes each sector on the duplicate disk. This option can back up normal, or "almost normal", disks more reliably, and can handle a few protection schemes more readily than MANUAL BIT COPY. However, MANUAL SECTOR COPY is not designed to copy disks whose formatting differs too greatly from DOS sectors.

After selecting MANUAL SECTOR COPY, you need to tell Copy II Plus which drives to use and what tracks to copy:

```
ORIGINAL  DRIVE:
DUPLICATE DRIVE:

ENTER START TRACK:
ENTER END   TRACK:
 TRACK INCREMENT:
```

(And again, NUMBER OF SIDES: for 3.5" Bit Copy.)

For start track, end track, and track increment, you should only use whole track numbers, not half-tracks or quarter-tracks.

You'll then see:

```
USING SECTOR COPY
```

followed by the usual — INSERT DISKETTES — display. If you need to change any parameters before starting the sector copy, press [/] to change them now. Otherwise, insert your disks into the appropriate drives, then press [RETURN] to start the copy.

Note: When you use MANUAL BIT COPY or MANUAL SECTOR COPY, Copy II Plus does not change the parameters back to their original values. If you need to copy more than one range of tracks, the parameters you set for the first range will still be set unless you change them again. However, when you copy a program with AUTO COPY or PARTIAL AUTO COPY, Copy II Plus restores all parameters to their original values before it reads the new parameter settings from the parameter entry. That way, you can AUTO COPY several disks in a row without worrying about the previous parameter settings. The entry you choose will also automatically select either Bit Copy or Sector Copy for you.

If you want to restore all parameters from MANUAL BIT COPY or MANUAL SECTOR COPY, press [/] and select to change parameter $FF. This is a special parameter. Instead of asking CHANGE WHAT VALUE, it will display:

```
        - RESTORE PARAMETERS -

ARE YOU SURE? Y
```

Press [Y] or [RETURN] to restore all parameters to their original values.

NIBBLE EDITOR

You can use the NIBBLE EDITOR option to see the actual bytes stored on any track of the disk. This can be invaluable for learning about disk formatting, or helping to determine what protection scheme or schemes a disk uses. When you select the NIBBLE EDITOR option, you can view the track data, but you can't change it. Later we'll explain how to use the nibble editor from within a disk copy so that you can make changes to the disk itself. (By the way, it's called a nibble editor because the disk bytes are sometimes referred to as "nibbles".)

When you select the NIBBLE EDITOR option from the main Bit Copy menu, you'll be asked:

```
ORIGINAL  DRIVE:

ENTER START TRACK:
ENTER END   TRACK:
TRACK INCREMENT:

SYNCHRONIZE TRACKS?
```

It doesn't ask for a duplicate drive since you're not doing any copying. It does ask for start track, end track, and track increment so that you can nibble edit several tracks in a row if you want. If you answer Yes to the SYNCHRONIZE TRACKS question, it will "align" the track immediately before reading the data. (See below.)

After answering the above questions, you'll get the usual — INSERT DISKETTES — prompt. Insert the disk you want to examine into the appropriate drive and press [RETURN]. The disk will whir as the track is read into the memory buffer, or track buffer.

The memory buffer is simply a large portion of the Apple's memory set aside for storing the bytes that are read in from the track. The nibble editor reads two or three revolutions of the track into this buffer. In most cases it starts reading from any arbitrary point on the circular track. This means if you read the same track twice, the data will probably not be in the same place in the buffer each time.

If you selected SYNCHRONIZED TRACKS, then the nibble editor will seek and synchronize itself to a point on another track (usually track 0), then immediately seek back and begin reading. If you read the same track twice using SYNCHRONIZED TRACKS, the data will appear within a few bytes of the same place each time. (This is also the same synchronizing that's done during a bit copy.)

You'll then see a display similar to:

```
   COPY ][ PLUS BIT COPY PROGRAM 8.n
(C) 1982-7 CENTRAL POINT SOFTWARE, INC.
----------------------------------------

TRACK: 00  START: 5F00  LENGTH: 44FF

5EE0: 80 80 80 80 80 80 80 80   VIEW
5EE8: 80 80 80 80 80 80 80 80
5EF0: 80 80 80 80 80 80 80 80
5EF8: 80 80 80 80 80 80 80 80
5F00: 9E AE AE DC E6 AF AB B9  <-5F00
5F08: F5 E6 E6 DF DA F6 CF F9
5F10: D3 DD FE EF F3 B5 F6 CF
5F18: F7 B5 F3 CE D7 FC CE EA
5F20: DE 96 FA BE F3 CE F7 B5

----------------------------------------

  A  TO ANALYZE DATA  ESC TO QUIT

  ?  FOR HELP SCREEN  /  CHANGE PARMS

  Q  FOR NEXT TRACK   SPACE TO RE-READ
```


The first line of the nibble editor display indicates what track you are currently editing, its start address in Apple memory, and its length. Since no analysis has been done yet, this is the start address and length of the entire buffer, not of the track data. Beneath this is the actual track image. It is shown as the Apple memory address followed by 8 hexadecimal bytes (or 16 hexadecimal bytes in the 3.5" Bit Copy) per line. The word "VIEW" to the right lets you know you are in VIEW mode (there is also a CHANGE mode, described below), and you can scroll through the track buffer. The address at the right marked by "<-" is the actual memory address of the byte that's under the flashing cursor. Sync bytes are shown in inverse.

Several options are displayed in the bottom window. You can ask Copy II Plus to perform its track analysis by pressing [A]. The track analysis routines, using the current parameter settings, determine the start and end of the track data, then move the

cursor to the track start and change the START and LENGTH values at the top to reflect the track size rather than the memory buffer size.

Pressing [Q] will quit this edit and move on to the next track. [ESC] will exit the editor and return you to the main Bit Copy menu, and [SPACE] will re–read the track and position the cursor back to the beginning of the memory buffer. [/] operates just as it does when copying disks, allowing you to change parameters.

If you press [?], you will be presented with a help screen which shows you what other commands are available from the nibble editor.

The cursor moving commands are different between the 5.25" and 3.5" Bit Copy programs, but they are quite straightforward, and let you move anywhere within the track buffer with a minimum of effort. The help screen shows you those commands. (The 5.25" Bit Copy must work with the more limited Apple II Plus keyboards, hence the difference.)

[CTRL–B] and [CTRL–E] can be used to establish a new track beginning or track end at the current cursor position. The START and LENGTH values will change, so you can use these commands to calculate the "distance" (in bytes) between any two bytes in the buffer.

[C] allows you to change nibbles, and you will notice the "VIEW" status becomes "CHANGE" when [C] is pressed. You may then enter any string of hex bytes separated by spaces and they will be placed at the current cursor position.

[F] allows you to find a string of bytes in the buffer. You will see the prompt "FIND" appear in the lower right of the nibble edit display. You can enter any 1 to 3 byte sequence for the editor to find. Spaces are optional. If the string is found, the cursor is moved to the first byte of the string. If it is not found, the cursor

is moved to the end of the track buffer. You can also enter the single byte "00" to find the next sync byte in the buffer. Pressing [R] will repeat the find command for the last specified string.

[S] will toggle the byte at the current cursor position between sync (shown in inverse) and standard (normal), converting standard bytes to sync, and sync bytes to standard.

[P] allows you to print a track. It will start printing at the current cursor location and extend to the end of the buffer if no analysis has been done, or to the track end if analysis has been performed. The printer slot number and page length are Copy II Plus parameters (parameter numbers $48 and $4A) and may be changed at any time. The sync bytes in the buffer are printed with their high bits cleared. (For example, a sync $FF will be printed as a $7F.)

When examining a track with the nibble editor, using [/], [SPACE], and [A] in sequence allows you to view a track, make any parameter changes you wish, then re-read and analyze the track using the new parameters. This analysis is the same that Copy II Plus uses when copying a disk.

As mentioned earlier, if you choose the NIBBLE EDITOR option from the main Bit Copy menu, you can read the track and make changes to it in memory, but you can't write those changes back to the disk. If you do want to make changes to the disk itself, there is a different method for entering the nibble editor. Choose MANUAL BIT COPY, selecting the tracks you want to edit, then set parameter $0B to 2. This tells Copy II Plus to "copy with nibble editor entry". It will read a track from the original disk, then pop you into the nibble editor so you can edit that track. When you're finished editing, press [Q] to quit out of the editor. It will resume the copy process, writing the edited track to the duplicate disk. (If you want to read and write the same disk, then set both the original and duplicate drives to the same drive number.)

When using the editor from MANUAL BIT COPY rather than the NIBBLE EDITOR option, the [A] to analyze, [/] to change parameters, and [SPACE] to re-read commands are not available. Copy II Plus has already set parameters and read and analyzed the track as part of the copying process before entering the nibble editor.

(If you're interested in better understanding disk formatting and protection schemes, we suggest you begin by using the nibble editor to examine a standard DOS or ProDOS disk, identifying the various address and data fields described in Appendix A. Then try examining and comparing the formats of various protected disks.)

SECTOR EDITOR

The Sector Editor allows you to directly view and change the data on any sector of the disk. This is handy for people interested in poking around files or Track/Sector Lists, etc. to learn more or to fix problems. It can also be used with the Bit Copy options for copying certain protected disks. You should use care when working with the Sector Editor, to avoid accidentally erasing or modifying important data on the disk.

A good knowledge of hexadecimal, bytes, and ASCII is helpful when using the Sector Editor. Appendix A is a reference on disks, tracks, sectors, and blocks. Please refer to Appendix A if you need more information about these things.

(Note: The Sector Editor in the 5.25" Bit Copy program is designed to work only with 5.25" floppy drives; similarly, the Sector Editor in the 3.5" Bit Copy is designed to deal only with 3.5" drives. These options use customized methods for properly dealing with strange sector formats and copy-protection schemes (see "Patch" option below), which isn' the same as the "generic" methods required by hard disks and RAMdisks.)

To use the Sector Editor, choose the SECTOR EDITOR option with the arrow keys and [RETURN]. You'll be asked for "ORIGINAL DRIVE:". Enter the drive number you want to use. The Sector Editor display will appear next, with the sector buffer cleared to zeros. This display will be explained shortly.

Notice the help prompt at the bottom of the screen:

```
[?]-HELP SCREEN
```

Press [?] to see the help screen, which shows what commands are available. (This help screen, like the NIBBLE EDITOR help screen, is slightly different between the 5.25" and 3.5" Bit Copy programs.) Press [RETURN] to go back to the Sector Editor buffer display.

There are some important differences between sectors on 5.25" disks and sectors on 3.5" disks:

Reading Sectors (5.25" Bit Copy)

To read a sector on a 5.25" disk, press [R] for Read. You will be prompted to enter the track and sector numbers of the sector you want to read. Enter the hexadecimal track number and press [RETURN], then enter the hex sector number and press [RETURN]. (All numbers used in the Sector Editor are hexadecimal.) An invalid character or an invalid number will cause the speaker to beep. After you enter the track and sector numbers, the sector will be read from the disk into the buffer.

As an example, insert the Copy II Plus disk into the drive and select to read track $00, sector $B. (This sector is part of the disk's directory information.)

Press [R] for Read,
Type "0" for the track number,
Press [RETURN],

Type "B" for the sector number,
Press [RETURN].

The disk will whir and you should see a display similar to:

```
SECTOR EDITOR                          DRIVE 1

00- 00 00 03 00 FE 43 4F 50  ....~COP
08- 59 49 49 59 4C 55 53 00  YIIPLUS.
10- 00 00 00 00 00 00 00 00  ........
18- 00 00 00 00 6F AB 00 00  ....o+..
20- 00 00 C3 27 0D 06 00 06  ..C'....
28- 00 18 01 26 50 52 4F 44  ...&PROD
30- 4F 53 00 00 00 00 00 00  OS......
38- 00 00 00 FF 08 00 1E 00  ...#....
40- 00 3A 00 6F AB 00 00 00  .:.o+...
48- 00 21 00 20 11 A9 00 00  .!. .)..
50- 02 00 2B 55 54 49 4C 2E  ..+UTIL.
58- 53 59 53 54 45 4D 00 00  SYSTEM..
60- 00 00 FF 26 00 36 00 F0  ..#&.6.p
68- 68 00 6F AB 00 00 00 00  h.o+....
70- E3 00 20 5C AB 00 00 02  c. \+...
78- 00 2E 42 49 54 43 4F 50  ..BITCOP
80- 59 2E 53 59 53 54 45 4D  Y.SYSTEM


TRACK $00, SECTOR $0B     DOS 3.3

[?]-HELP SCREEN
```

The track and sector number you just read is shown at the bottom of the screen, along with the DOS "patched" option, which in this example is "DOS 3.3". The Patch option is explained later.

Seventeen lines of the sector are displayed at a time, consisting of a hex "address" followed by a dash, then 8 hex data bytes (each byte is a two digit hexadecimal number), then the same 8 bytes as ASCII characters on the right. The "double cursor" appears in inverse over both the first hex value and the first character. The characters on the right may or may not make sense. (In the example above, the filenames for this disk can be

read on the right, along with other values that were never intended to be printed as characters.)

To understand the addresses on the left, think of the data bytes numbered from $00 as the first byte of the sector to $FF as the last byte. The top line shows the first 8 bytes, bytes $00 through $07; the next line shows bytes $08 through $0F; the next shows bytes $10 through $17, etc. The address number before the dash tells you how many bytes into the sector each line is ($00–, $08–, $10–, etc.). The address number of a byte is not the same as the value of that bytes. In the example, the addresses of the first four bytes on the second line are $08, $09, $0A, and $0B. The values of those bytes are $59, $49, $49, and $59.

Reading Sectors (3.5" Bit Copy)

In addition to the 512 bytes of data, each sector on a 3.5" disk also contains 12 bytes of "tag" information. These bytes are used only on Macintosh–format disks, and are always zero (00) on Apple–format disks. The Sector Editor lets you see the values of the tag bytes at the top of the Sector Editor display:

```
SECTOR EDITOR                                                DRIVE 1

TAGS- 00 00 00 00 00 00 00 00 - 00 00 00 00                  .......

 000- xx xx xx xx xx xx xx xx - xx xx xx xx xx xx xx xx  .......
 010- xx xx xx xx xx xx xx xx - xx xx xx xx xx xx xx xx  .......
 020- xx xx xx xx xx xx xx xx - xx xx xx xx xx xx xx xx  .......
   etc.
```

When you press R for Read or W for Write, you can specify which sector to read or write by either TRACK, SECTOR, and SIDE, or by BLOCK number. (See Appendix A for more information about these terms.)

When you first enter the Sector Editor option, at the bottom of the screen you'll see BLOCK on the left, then TRACK, SECTOR, and SIDE to the right. If you press the M key (to

change "Modes"), TRACK, SECTOR, and SIDE will appear on the left, with BLOCK to the RIGHT. Pressing M again will change the "Mode" back to BLOCK on the left.

To read a sector, press R. Copy II Plus will ask for whichever option is listed first, on the left. If BLOCK is to the left, you'll be asked to enter a block number. Type in the desired block number (in hexadecimal) and press RETURN. The corresponding track, sector, and side numbers will be filled in for you and that sector (block) will be read.

If TRACK, SECTOR, and SIDE are to the left of BLOCK when you press R to read, then you'll be asked to enter track, sector, and side numbers. Type in the three numbers (in hexadecimal), pressing RETURN each time. When all three numbers are entered, the corresponding block number will be filled in for you and that sector (block) will be read.

After you've read a block, the block number is shown in decimal (in parentheses) after the hexadecimal number (with a $ dollar sign).

Moving the Cursor

The 5.25" Bit Copy uses the keys [I], [J], [K], and [M] for moving the cursor through the buffer. (Notice that these four keys make a diamond pattern on your keyboard. This will help you remember which direction each key goes.) The 3.5" Bit Copy uses the four arrow keys. The buffer display will scroll up or down to keep the cursor on the screen. [B] moves the cursor directly to the beginning of the buffer; [E] moves the cursor to the end.

You can also move the cursor to any address in the sector or find out what address the cursor is currently at. Press [A] for Address. You'll see:

```
ENTER ADDRESS: nn
```

with an address number displayed. This address is simply how many bytes into the sector the cursor is. If you don't want to move the cursor, just press [RETURN]. If you want to move to a new address, type the new address number, then press [RETURN]. The cursor will immediately jump to the new position in the buffer.

Reading Again

If you want to read a different sector from the disk, you can press [R] again, and enter new numbers. You can also read the next higher numbered sector on the disk by pressing [+], or read the previous sector by pressing [–].

Changing Bytes

You can change the data in the sector buffer by entering either new hex values or new text characters. To enter hex values, move the cursor to the appropriate place and press [H] for Hex. The cursor over the hexadecimal value will change to a blinking underline. Now enter the new value over the old. Pressing [RETURN] will advance you to the next byte, and pressing [ESC] will take you out of hex entry.

To enter characters, position the cursor and press [T] for Text. The cursor over the text character will change to a blinking underline. Typing new characters will enter those characters into the buffer and advance the cursor. Press [ESC] to finish text entry. Note: While entering text, any control characters typed (including the arrow keys and [RETURN] but not including [ESC]) will be placed directly into the buffer.

Writing

To write a sector back to the disk, press [W] for Write. You will again be prompted for track and sector (or block) numbers. If you want to write back to the same sector, just press [RETURN] to accept the same values. If you want to write to a different sector, enter new values. The disk will whir as the sector is written.

How to Edit a Sector

With the options presented so far, you can do most sector editing. Editing a sector consists of reading the sector, changing the appropriate bytes, then writing the changed sector back to the disk. Here's a step-by-step method for making a change to a sector on the disk:

1. Do not sector edit a commercial disk! Make a copy of the disk first, then sector edit the copy.

2. Choose the SECTOR EDITOR option and select the desired drive number. Insert the disk you want to edit.

3. Press [R] for Read, and enter the track and sector or block numbers of the sector you want to edit. Copy II Plus will read the sector into the memory buffer.

4. Position the cursor using the appropriate keys to the address where you want to make changes.

5. Press [H] and enter new hex values, or press [T] and type new text characters, to replace the old. If you're entering several hex values in a row, you can press [RETURN] after entering each byte to advance to the next position. Press [ESC] to finish the entry.

6. Press [W] for Write, to write this changed sector back to the disk.

Disassembly

The Sector Editor can disassemble and list any 6502 machine language code that may be in a sector. Position the cursor on the first byte you want to disassemble and press [L] for List Disassembly. The sector buffer display will be replaced by 20 lines of disassembled code. The cursor also advances through the sector by the number of bytes disassembled. Press [L] to disassemble another 20 lines, or [RETURN] to go back to the buffer display.

Printer Dumps

Using the Printer Dump option, you can print either the buffer display or a disassembly listing. This option will normally print through slot 1. (If your printer card is in another slot, first change the value of parameter $48 to the slot number.)

To print the sector buffer, press [D] for printer Dump. All 32 lines (256 bytes) of the sector will be printed. To print a disassembly listing, first press [L] to disassemble the code on the screen, then press [D]. Twenty lines of disassembly listing will be printed. Press either [L] or [D] to print another 20 lines. Press [RETURN] to stop printing and return to the screen buffer display.

Scan for Bytes

Another feature in the Sector Editor is the ability to scan for a pattern of bytes anywhere on the disk or within a file. If you haven't read any sectors yet, this option will scan the entire disk. If you have read a sector, it will scan from the current position to the end of the disk.

To scan for Bytes, Press [S] for Scan. You can enter the bytes to scan for as either hex values or text characters. A question will appear:

```
SCAN FOR [H]EX OR [T]EXT?
```

Type [H] or [T]. If you select [H], it will then ask "ENTER HEX:". Type in the hex values (one or two digits each) that you want to scan for, separated by spaces. If you select [T], it will ask "ENTER TEXT:". Type in the characters you want to scan for.

You can use the left–arrow key to go back and correct mistakes, and the right–arrow key go over values already typed. Press [RETURN].

The program will then rapidly scan the disk, looking for the bytes you specified. If it finds them, it will stop and display that sector, with the cursor over the last byte of the pattern. If it can't find the pattern, it will say "BYTES NOT FOUND".

If you want to scan for another occurrence of the same pattern, just press [S], then press [RETURN] twice to accept the previous answers you gave to the two questions. The program will continue scanning.

Patch

Another Sector Editor option is [P], for Patch Read/Write Routines. Normally the Sector Editor can read only standard formatted disks. Some protected programs use a slightly modified sector format, so that the disk cannot be read with a normal DOS. The Patch option lets you read or write these changed sectors. Other protected disks might use a very different disk format that does not contain "sectors" at all! The Sector Editor cannot read these disks.

We recommend that you use the Patch option only if (1) you're sector editing a backup of a commercial program and you have

instructions on what Patch option to use, or (2) you're familiar with disk and sector formatting. Appendices A and B provide information about sector formats.

The following examples are for the 5.25" Sector Editor. The differences in the 3.5" Sector Editor Patch option are minor, and are discussed below.

5.25" Bit Copy: To show how the patch option works, remove the disks from your drives (we're being safe here!) and press [P] for Patch. A screen similar to the following will appear:


```
SECTOR EDITOR PATCHER                DRIVE 1

DOS 3.3
DOS 3.3 PATCHED
DOS 3.2
DOS 3.2 PATCHED
CUSTOM
---------------------------------------
DOS 3.3

                  ADDRESS   DATA

        PROLOG: D5 AA 96  D5 AA AD
 WANTED EPILOG: DE AA     DE AA EB FF FF
   READ EPILOG= DE AA     DE AA EB FF ED
CHECK CHECKSUM? YES       YES
  CHECK EPILOG? YES       YES
   CHECK TRACK? YES
           DATA ENCODING: 6&2
           CHECKSUM SEED: 00
         CHECKSUM RESULT= 00

USE ARROW KEYS & [RETURN] TO SELECT
PATCH OPTION, [ESC]-EXIT
```

The menu at the top of the screen lets you select what type of sector you can read or write. You can select normal DOS 3.3 (which is actually for any normal 16-sector disk, including ProDOS or DOS 3.3) or 3.2 (for 13-sector disks), or DOS 3.3 PATCHED or DOS 3.2 PATCHED. The "PATCHED" items

adjust the Copy II Plus read/write routines enough to read many protected disks, while still reading normal sectors almost as reliably.

Right below the dashed line, it shows which patch option is currently selected (in the example, DOS 3.3). The rest of the display shows the internal values and settings that make up that particular patch option.

If you want to select another patch option, use the arrow keys to display that option in inverse, then press [RETURN]. The display below the dashed line will change to reflect the new patch option. For this example, select "DOS 3.2 PATCHED". Notice that it now says "DOS 3.2 PATCHED" below the dashed line.

Press [ESC] to go back to the Sector Editor screen. Beside the track and sector numbers, it now shows "DOS 3.2 PATCHED", which is the new patch option you just selected.

3.5" Bit Copy: The Patching options for the 3.5" Sector Editor are:

NORMAL
PATCHED
CUSTOM

The NORMAL option reads and writes sectors on 3.5" disks with all the error checking that would normally be done by a program. The PATCHED option adjusts the Copy II Plus read/write routines enough to read many protected disks, while still reading normal sectors almost as reliably. (The CUSTOM option is described below.)

Note: When you leave the Sector Editor, if you use MANUAL SECTOR COPY next, the read/write routines will stay patched. If you choose AUTO COPY, then the AUTO COPY routines will always set things back to normal before following the instructions in the parameter entry.

How to Set "Patched" Routines

1. Press [P].

2. Press the arrow keys to highlight the option you want.

3. Press [RETURN]. The display below the dashed line will change to show the new option.

4. Press [ESC] to go back to the Sector Editor screen. You can now read or write sectors using the new patch option.

Custom Patching

The last option in the Patch menu is CUSTOM. Custom patching lets you tailor the read/write routines to access a wide variety of possible protected-sector formats. A good technical understanding of sector address and data fields is essential for what follows.

The sector "parameters" on the screen are used by Copy II Plus when either reading or writing sectors. The READ EPILOG and CHECKSUM RESULT fields give you information about the sector that was last read. They're blank if you haven't read any sector yet. You can change all of the other fields to determine what kind of sector to read.

When you select CUSTOM from the patch menu, an inverse cursor appears over one of the data values. To move the inverse cursor forward through the list of values, you can press [RETURN], [SPACE], or the right-arrow key. To move backwards, press the left-arrow key. When the cursor is over any hex value, you can type a new value to change it. If the cursor is over a YES-NO response, typing [Y] will change it to YES and [N] to NO. If the cursor is at the DATA ENCODING question

(5.25" only), you can type [5] to use 5&3 encoding, or [6] to use 6&2 encoding. Press [ESC] to leave CUSTOM patching and go back to the patch menu. Press [ESC] again if you want to return to the Sector Edit buffer display.

When reading, both address and data prologs must match the PROLOG fields. Volume is ignored. Track number is "partially" ignored if you answer NO to the CHECK TRACK question. That is, Copy II Plus will seek to the proper track, but will not re-seek if the track number in the address field differs. Sector number must match. Address and data field checksums and epilogs can be checked or ignored. If epilogs are checked, then the first two bytes of each epilog must match the first two bytes in the WANTED EPILOG fields. The actual epilog bytes read appear in the READ EPILOG fields. The CHECKSUM SEED value is the starting value used when exclusive-ORing the data field into memory. It can range from $00 to $3F for 6&2 encoding or $00 to $1F for 5&3 encoding. For normal sectors, this byte should be $00 to read the data correctly. The data CHECKSUM RESULT is formed by exclusive-ORing the running data checksum with the checksum byte on disk. If this byte is nonzero, the data checksum test fails. This means either the sector was written with a different CHECKSUM SEED value, or there's an error in the data field, or the data checksum byte at the end of the data field is wrong. (The 3.5" Patch menu includes three checksum seeds and three checksum result

When reading a sector, Copy II Plus tries to find an address field and data field pair on the track that passes all the tests. If it fails after many tries, it gives up and prints an "I/O ERROR" message. You can sometimes find out how far it got by checking the Patch display after you get the error. If it can find a correct address prolog, it will finish reading the address field and the address READ EPILOG values will be filled in. If it finds a correct data prolog, it will read the rest of the data field and the data READ EPILOG and CHECKSUM RESULT values will be filled in.

When writing, it must first read the appropriate address field, then write a new data field over the old. The address field parameters behave as described above. The new data field prolog is written using the data PROLOG bytes. The data is exclusive-OR'ed and written using CHECKSUM SEED as a starting value. This should be $00 to write normal sectors. If the data CHECK EPILOG field is set to YES, then the WANTED EPILOG bytes will be written as the data epilog. If CHECK EPILOG is set to NO, then the READ EPILOG bytes are used. This allows the routines to automatically write the same epilog it read. It writes 5 epilog bytes (rather than 2 or 3) because a few protected disks check for these extra bytes.

HI-RES DISK SCAN

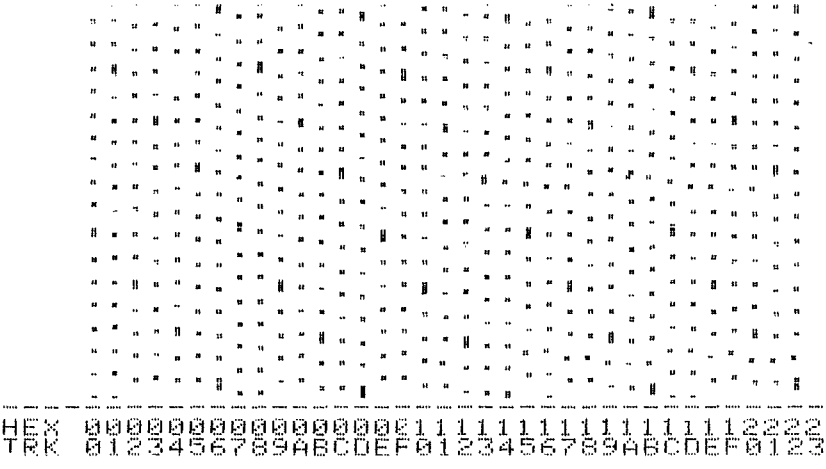
The HI-RES DISK SCAN option is a quick graphical tool to help you determine which tracks or half-tracks on a disk contain useful data, and which tracks are "blank". It does this by showing you the general pattern of sync bytes and invalid bytes on any tracks you specify.

HI-RES DISK SCAN reads each track into the track buffer, then divides it into groups of 41 bytes each. If there are any invalid bytes or sync bytes in the group, Copy II Plus plots a dot on the high-resolution graphics screen. If there are no invalid or sync bytes in the group, it leaves that point black. The dots for each track are plotted in a vertical line, from top to bottom of the screen.

To use HI-RES DISK SCAN, choose the option from the main Bit Copy menu, then answer the questions concerning drive, track range, and synchronized tracks. Insert the disk you want to scan, then press [RETURN]. The DISK SCAN screen will appear, with the hexadecimal track numbers ($00 to $23 for 5.25" disks, $00 to $4F for 3.5" disks) at the bottom of the screen. Vertical lines or dots will appear above each track

number as the track is scanned. Press [ESC] if you want to exit out before it's finished, or press any key to exit when it's done.

Here is a picture of a 5.25" DISK SCAN of a normal DOS 3.3 16-sector disk.



Each track is plotted in a vertical line over the track number. Any normal 16-sector disk will produce a display similar to this. The dots are the sync fields between the sectors. The short stripe on each track is the longer sync field at the start of the track.

If you scan a normal DOS disk on the (unwritten) half-tracks, you'll see irregular patterns of stripes and dots. This is caused by the drive trying to read bytes from the whole tracks on either side of the half-track, leaning toward one track or the other.

(Note: The patterns will not line up from one track to the next. The timing used when stepping from track to track is not the

same as when the disk was written, so each pattern begins at a different point around the circular track.)

If you scan a disk that has never been formatted or written to, you will see a solid stripe for each track. This is because an unformatted disk contains many invalid bytes around each track, which show up as white. Unused tracks on a protected disk will also appear as white stripes. (The patterns for a 3.5" disk are similar in meaning to those of a 5.25" disk. There are simply more stripes, because the 3.5" disk has more tracks.)

The HI-RES DISK SCAN option provides you with a quick way to see some of the peculiarities of a protected disk. You can use DISK SCAN to help locate the more "interesting" tracks, then use the nibble editor to examine those tracks more closely.

Chapter Five – Parameter Entries

This chapter describes the Bit Copy options that allow you to create and edit your own AUTO COPY parameter entries, and add these to the list of parameter entries already on the Copy II Plus disk. If you've found out from a friend (or read in a magazine or computer newsletter, etc.) what the Copy II Plus parameters for backing up a program are, then you might want to make a new parameter entry in the list for copying that program. You'll need to know how to create a new entry, type in the special parameter values, and save the new entry to disk. If you've discovered how to back up a program yourself, you'll also need to understand what the individual parameter values mean, so you can make an entry that does what you want.

The individual items in a parameter entry are described first, then the Bit Copy options for changing the parameter entries are explained. If you just want to type in an entry that's been provided for you, you might want to skim the following material, then pick up again below under LOAD PARM ENTRY.

Each parameter entry is a set of special instructions which Copy II Plus can use when backing up a particular program with AUTO COPY. The instructions tell Copy II Plus how to set start and end track, track increment, any parameter changes, etc., before copying the disk. These parameter changes are the same as those used in MANUAL BIT COPY and MANUAL SECTOR COPY, which were described in the last chapter.

Here are the main instructions used in parameter entries. Each instruction is described first, then followed by short examples where appropriate.

Txx–Tyy	Copy from track xx to track yy. In other
words,	select a START TRACK of xx and an END
TRACK	of yy.
T0–T22	copies from track $0 to track $22.
T11–T1B	copies from track $11 to track $1B.

T1.5–T7.5	copies from track 1.5 to track 7.5. (These are half–tracks.)
T3.75–TE.75	copies from track 3.75 to track E.75 (quarter–tracks).
T4–T5	copies tracks $4 and $5.
Txx	Copy only track xx. Set both START TRACK
and	END TRACK to xx.
T0	copies only track $0.
T21	copies only track $21.
STEP zz	Select a track increment of zz.
STEP 2	selects a track increment of 2 (which would copy every other track).
STEP 1.5	selects a track increment of 1.5.
SYNC	Answer Yes to the SYNCHRONIZE TRACKS question.
KEEP	Answer Yes to the KEEP TRACK LENGTH question.
xx=yy	Set parameter number xx to value yy.
3E=2	sets parameter $3E to 2.
10=97	sets parameter $10 to $97.
RESTORE	Restore all parameters to their original values. This command should always be on a line by itself.
SECTOR COPY	Do a sector copy rather than a bit copy. If no tracks are specified (see below), then it copies the entire disk. If tracks are listed, it only sector copies those tracks.
"COMMENT"	Any comments in the parameter entry should be in quotes and on separate lines. The comments will be displayed on the screen

during copying. You can have more than one line of comments, but each line should be enclosed in quotes.

The instructions that do a copy need to be separated by commas. Here are a few examples of instructions alone or combined together:

```
T0
T0-T22
TA-TE, SYNC
T0-T22, KEEP
T4-T5, SYNC, KEEP
T0-T8, STEP 2
T1.5-T7.5, STEP 1.5
T0, 3E=2
T2-T22, E=D4, F=AB, 10=97
SECTOR COPY
T0-T3, SECTOR COPY
T0-T3, SECTOR COPY, 57=D4
```

Remember that some protected disks use different protection schemes on different tracks of the disk. These disks often require several "passes" through the bit copy, each pass selecting a different track range and setting different parameters. When Copy II Plus reads a parameter entry to copy a disk, it reads all of the instructions from one line of the entry, sets the appropriate track numbers, parameters, etc., then does the copy. Then it reads the next line of the entry to do the next pass (if there is one). When creating the entry, you need to remember that all the instructions for one pass should be together on a line, and different passes should be on different lines.

Here is an example of a multi-pass parameter entry:

T0	First copy just track 0, no parameter changes.
T1.5-T7.5, 3E=2	Then copy half-tracks 1.5 to 7.5, after setting parameter $3E to $2.

T11–T21, SECTOR COPY	Then sector copy tracks $11 to $21.
T22, KEEP, 9=1	Lastly copy track $22, keeping track length (do nibble counting), after setting parameter 9 to 1.

3.5" Bit Copy only: If a track ("T5") or a range of tracks ("T7–T3E") is specified, Bit Copy will ordinarily copy both sides of each track. If you add a SIDE number (for example, "T5, SIDE 1"), it will copy just the desired side of the track or tracks.

The best examples can be found in the parameter entries stored on the Copy II Plus disk. We'll explain shortly how you can load and see these parameter entries.

Sector Edit Parameters

The Bit Copy program can also do automatic sector editing to the duplicate drive, controlled by a parameter entry with AUTO COPY. Sector editing is a novel method used to help back up certain protected disks.

On some protected disks, most of the program is stored using fairly normal DOS–type sectors, but one or two tracks contain special marks which a bit copy program may have trouble duplicating. When the program is loaded, it looks for these special marks on the disk. If it doesn't find them, it "knows" that this is a copy and not the original disk, and will refuse to run.

The sector edit approach is to actually modify part of the program stored on the duplicate disk so that when it boots, it simply ignores the fact that the marks are absent. The modification can either remove the protection check, or ignore the results of the check after the test has been done. Determining what kind of change to make to a specific disk is usually a major programming task, and the designers of copy-protection schemes often try to

make it as difficult as possible! If you already know what needs to be changed, though, it's fairly easy to make the change. (The SECTOR EDITOR option lets you make changes by hand.)

If an AUTO COPY parameter entry calls for sector editing, Copy II Plus will automatically do the sector edit to the duplicate disk. The only time you need to be aware of this is if you want to create your own parameter entries that include sector editing.

The sector edit instructions need to specify: which track and sector is to be modified, whether it is a DOS 3.3 or 3.2 type sector (for 5.25" disks), if the read/write routines should be "patched" (see the SECTOR EDITOR section in Chapter Four for a description of "patched"), any other parameters that may need to be set (for "custom" patching), and lastly the addresses in the sector to be changed along with their new values. Here, in the correct order, are the parameter entry instructions needed to do sector editing:

SECTOR EDIT,	This starts the sector edit.
TRACK xx,	Track number,
SECTOR yy,	Sector number,
DOS 3.n,	*5.25" disks:* DOS 3.3 for 16 sector disks, DOS 3.2 for 13 sector disks,
SIDE z	*3.5" disks:* Side number,
or BLOCK zzzz	*3.5" disks:* Block number (alternate
to	track, sector, side)
(optional) PATCHED,	PATCHED option if desired,
(optional parameter changes),	Any other parameter changes,
aa:dd,	The position (address) in the sector to change, and the data to change it to,
aa:dd/dd/dd	Changes to adjacent bytes in the
sector.	

Here are a few examples to clarify this:

5.25" Bit Copy:

```
SECTOR EDIT, TRACK 0, SECTOR 8, DOS 3.3, A0:60
```

This example edits the sector at track 0, sector 8, which is a DOS 3.3–type (16–sector format) sector. The byte at address $A0 is changed to a $60, then the sector is written back to the disk.

```
SECTOR EDIT, TRACK 22, SECTOR 1, DOS 3.2, PATCHED,
59=97, 14:00, D1:2F/AF/32
```

This edits track $22, sector 1 as a DOS 3.2–type (13–sector format) sector, using "patched" read/write routines. Parameter $59 is set to $97. The byte at address $14 is changed to a $00, then the three bytes starting at address $D1 are changed to $2F, $AF, and $32.

3.5" Bit Copy:

```
SECTOR EDIT, TRACK 4F, SECTOR 7, SIDE 2, 1FF:00
```

This example edits the sector at track $4F, sector 7, side 2. The byte at address $1FF is changed to a $00. (This happens to be the last byte on the last sector of a double–sided 3.5" disk.)

```
SECTOR EDIT, BLOCK 63F, PATCHED, 59=97, 14:00,
15D:2F,AF,32
```

This edits block $63F (specifying again, this time by block number, the last sector on the disk), using "patched" read/write routines. Parameter $59 is set to $97. The byte at address $14 is changed to a $00, then the three bytes starting at address $15D are changed to $2F, $AF, and $32.

If an I/O error occurs while Copy II Plus is trying to sector edit the duplicate disk, an error "7" will appear in the status display.

Sector editing should always be done to a copy of a commercial disk, never to the original!

LOAD PARM ENTRY

This Bit Copy option lets you select a parameter entry from the disk, load it into memory, then see and modify the instructions that make up the entry. When you choose LOAD PARM ENTRY, a new screen appears:

```
            LOAD PARM ENTRY

NAME:
```

Enter the name of the parameter entry you want to load, or press [RETURN] to see a list of all of the parameter entries. You can select the entry name from the list, just as in AUTO COPY. The disk will whir as the entry is loaded, then the "parameter entry edit screen" appears. Here is a sample edit screen:

```
NAME: RASTER BLASTER

BY: BUDGECO
----------------------------------------
T0
T5-T11, STEP 4, A=2, E=AD, F=DE, 55=3, 4
4=1, 45=10
T6-T12, STEP 4
T7.5-TF.5, STEP 4
T1.5-T3.5 STEP 2
"RETRY TRACK ZERO UNTIL BOOTS"



----------------------------------------
```

The first line shows the name of the parameter entry. The "BY" line shows the software publisher's name. (This line may be blank in some entries.) Below the dashed line are the bit copy instructions that make up the entry. Notice that the second instruction line was too long and wrapped around to the next line on the screen. That's okay.

You can make changes to the parameter entry in memory if you want. If you press [RETURN] twice, that will keep the same entry name and "BY" name. You can also type new names over the old. This is handy if you want to create a new parameter entry by editing an old one. The original entry on disk will remain unchanged.

The name and the BY line can be up to 29 characters long, and contain any characters except "*" and "_". After you enter a new entry name, a "*" will appear by the name. (Parameters on the Copy II Plus disk that were submitted by users all have a "*" by the name. Parameters that were tested and verified by Central Point Software do not have a "*".)

Once the cursor is down in the instruction area, it acts like a miniature word processor. Typing characters inserts those characters into the line.

5.25" Bit Copy: The left-arrow key deletes characters. To move the cursor, press [ESC]. The blinking underline cursor will change to a flashing plus-sign. Pressing [I], [J], [K], [M] will move the cursor up, left, right, down. (The diamond pattern these four keys make on the keyboard will help you remember which direction they move.) Press any other key to change back to a normal cursor. After you've pressed [ESC] to make the cursor a flashing plus-sign, you can also press [?] to see a help screen of "PARM ENTRY EDITOR COMMANDS".

3.5" Bit Copy: The [DELETE] key delete characters. Use the four arrow keys to move the cursor. You can also press [?] to see a help screen of "PARM ENTRY EDITOR COMMANDS".

For either Bit Copy, when you press [RETURN] to end a line or move the cursor to another line, Copy II Plus checks the line to make sure it contains only valid parameter entry instructions. If there is an error, Copy II Plus will print an error message at the bottom of the screen and leave the cursor on the line with the

error. Here are some examples of incorrect instructions with the error messages they produce:

```
T6-T5        END TRACK < START TRACK
```

The start track number needs to be less than the end track number.

```
TQ           BAD TRACK NUMBER
```

"Q" is not a valid track number.

```
XYZABC123    SYNTAX ERROR
```

Copy II Plus can't make sense of what you typed. It's not a valid parameter entry instruction.

You can also print the parameter entry on your printer. Press [CTRL–P] anytime the cursor is in the instruction area. Copy II Plus will display the printer slot number (slot 1, unless you change it) and ask you to press [RETURN] to print the entry.

When you want to quit out of parameter editing and go back to the Bit Copy menu, press [CTRL–Q] (5.25" Bit Copy) or [ESC] (3.5" Bit Copy).

EDIT PARM ENTRY

Whenever you use AUTO COPY, PARTIAL AUTO COPY, or LOAD PARM ENTRY, the parameter entry you last selected is stored in the computer, in case you want to use it again. With the EDIT PARM ENTRY option, you can look at or modify whatever parameter entry is currently stored in memory. When you select EDIT PARM ENTRY from the Bit Copy menu, Copy II Plus displays the parameter entry edit screen, the same one used in LOAD PARM ENTRY. As before, you can change the NAME and BY lines, or press [RETURN] to accept the current lines. Then you can use the editing keys to change the instructions

that make up the parameter entry. Press [CTRL-Q] (5.25" Bit Copy) or [ESC] (3.5" Bit Copy) to exit.

CREATE NEW PARM ENTRY

Choose this option when you want to create a new parameter entry from scratch.

Copy II Plus will show you a blank parameter entry edit screen with the cursor flashing on the NAME line. Type the name you want to give this new parameter entry. You must type at least one character for this field. Then fill in the BY line. This can be blank if you want. Now type in the copying instructions for the parameter entry, following the rules that were given earlier under "Parameter Entries" and "LOAD COPY". As before, press [CTRL-Q] or [ESC] to exit the editor.

If you create a new parameter entry, you can use AUTO COPY to test it out if you want, before saving the entry to disk.

SAVE PARM ENTRY

After you've made changes to a parameter entry or created your own parameter entry, choose SAVE PARM ENTRY if you want to to save it back to the disk to make it permanent. The disk will whir as Copy II Plus saves the parameter entry.

If there is already a parameter entry with that name stored on the disk, Copy II Plus will print:

```
ENTRY ALREADY EXISTS
        REPLACE IT?
```

Press [Y] or [RETURN] to replace the old entry with the new; press any other key if you don't want to save it.

Note: You should normally save parameter entries onto your work copy of Copy II Plus. The 5.25" parameter entries are recorded in two files on the disk, called PARM.KEY and PARM.DATA. The 3.5" entries are recorded in PARM35.KEY and PARM35.DATA. The corresponding Bit Copy program looks for these files when it saves an entry. If it can't find the files, then it creates them on the disk, then saves the parameter entry into them. This is handy if you want to store your own parameter entries onto another ProDOS disk. However, if you always want to save the entry onto the Copy II Plus disk, you need to be sure the disk is in the drive before you select SAVE PARM ENTRY.

Another note: The Copy II Plus 5.25" disk is nearly full, and has room for only a few more parameter entries. If you want to add a number of new parameter entries, do this: Make a copy of the Copy II Plus disk, then delete the file UTIL.SYSTEM from the copy. You'll then have a disk which boots directly to the Bit Copy program and has plenty of room for more parameter entries.

RENAME PARM ENTRY

Choose RENAME PARM ENTRY if you want to change the name of one of the parameter entries stored on disk. To choose which parameter entry to rename, you can either type in the old name or press [RETURN] and select the name from the entry list. Then Copy II Plus will ask for NEW NAME. Type the new entry name. Remember that this can be 1 to 29 characters long, and can include any printing character except for an asterisk or underline. When you press [RETURN], the disk will whir as Copy II Plus renames the entry.

DELETE PARM ENTRY

To delete a parameter entry from the entry list, choose DELETE PARM ENTRY, then type the name of the entry to delete or press

[RETURN] to choose from the parameter entry list. The entry will then be deleted.

Possible Parameter List Errors

If there is a problem when loading or saving a parameter entry, Copy II Plus will print an error message. Here is a summary of possible errors:

```
- WRITE PROTECT ERROR -

   PLEASE REMOVE
 WRITE PROTECT TAB
   FROM DISKETTE
```

This error will occur if you're trying to save, rename, or delete a parameter entry on the disk. Remove the write-protect tab from the disk and try again.

```
THE PARM ENTRIES ON
 THIS DISKETTE HAVE
   BEEN DESTROYED
```

This not-very-pleasant message means that the files that contain the parameter entries are somehow damaged. The parameter entry you requested cannot be loaded. You should make a new work copy from your original Copy II Plus disk, and use this new copy from now on.

```
- WRONG DISKETTE -

 PLEASE INSERT A
PARM FILE DISKETTE
```

Copy II Plus could not find the parameter entries on this disk. You probably have the wrong disk in the drive.

```
- I/O ERROR -

UNABLE TO LOAD OR
 SAVE PARM ENTRY
```

It can't read this disk. Either the information on the disk has been damaged, or the wrong disk is in the drive.

```
- DISKETTE FULL -

INSERT ANOTHER DISKETTE
   TO SAVE PARM ENTRY
```

There is no more room on this disk for saving parameter entries. You'll need to either delete any entries that you don't want, or start saving new entries onto another ProDOS disk. (See "SAVE PARM ENTRY" for more information.)

```
- PARM ENTRY DIRECTORY FULL -
```

Copy II Plus can keep track of up to 752 parameter entries on a disk. You just tried to save the 753rd entry. Delete the entries you don't want anymore, or start saving new entries onto another ProDOS disk.

```
- PARM ENTRY NOT FOUND -
```

You typed in a parameter entry name (or the first few letters of the entry name), and Copy II Plus couldn't find it in the list. You may have misspelled the name of the entry.

```
- ENTRY ALREADY EXISTS -
```

You're trying to rename a parameter entry, and the name you chose is already in the parameter entry list. You can't have two entries with the same name.

PRINT PARM FILE

The PRINT PARM FILE option will let you print out either the entire alphabetized list of parameter entries, or just a range of entries.

When you choose the PRINT PARM FILE option, Copy II Plus will ask:

```
STARTING ENTRY:

ENDING   ENTRY:
```

You can type in either an entire entry name or the first few letters of a name, or just press [RETURN]. Copy II Plus will print out every entry between (and including) the starting and ending entry you specify. For example, if you type "J" for a starting entry and "N" for an ending entry, it will print all of the entries that begin with J, K, L, M, or N. If you type "TRY" for both starting and ending entries, it will print all the parameter entries that begin with TRY.

If you press [RETURN] for the starting entry, the print-out will start with the first entry of the parameter list. If you press [RETURN] for the ending entry, it will print to the last entry of the parameter list. So to print the entire list, just press [RETURN] twice.

Appendix A – Disks and Disk Hardware

This appendix is included as a concise reference on disks and disk hardware. It explains disk formatting and storage, and most of the terms needed before exploring disk protection schemes. It is, however, a reference rather than a tutorial. For more complete information, and some useful examples for 5.25" disks, we suggest the book "Beneath Apple DOS" (or "Beneath Apple ProDOS") by Quality Software. Also, an appendix in Apple's DOS Programmer's manual describes DOS file formats, and "Understanding the Apple II" (also by Quality Software) describes the disk hardware in greater depth.

This reference assumes that you are familiar with computer concepts such as hexadecimal, binary, bytes, bits, and subroutines.

ProDOS, DOS 3.3, Blocks, Sectors

The Copy II Plus Utilities can work with disks created by two different Apple disk operating systems: DOS 3.3 and ProDOS. DOS 3.3, as originally supplied by Apple, is designed to work only with 35-track 5.25" floppy disks. ProDOS is capable of handling 5.25" disks, 3.5" disks, RAMdisks, hard disks, and any other disk device that follows certain conventions required by ProDOS. Both of these operating systems perform a number of tasks, including saving or writing files onto the disk, loading or reading files from the disk, and keeping track of where on the disk the files are stored.

Depending on what program is being run, DOS may need to access anywhere from one byte up to thousands of bytes from the disk at any one time. What is needed is a way to divide the information into manageable chunks.

ProDOS divides any disk device into numbered "blocks", each block containing 512 bytes of data. A disk might have, for

example, 280 blocks, numbered 0 to 279. For *5.25" disks only*, each block is actually made up of two "sectors" paired together, each sector containing 256 bytes of data (as described below). This pairing of sectors is handled by ProDOS's low-level floppy disk routines, however. The ProDOS file system is concerned only with whole blocks.

DOS 3.3, not having to handle other kinds of disk devices, divides a 5.25" disk directly into 256-byte sectors.

When a file is saved to disk, DOS 3.3 breaks the file into 256-byte chunks, looks on the disk for sectors that are not currently "in use", saves the chunks into the free sectors, makes a record on the disk of which sectors the file uses (so it can find the file later), and marks the sectors "in use". ProDOS is similar, except that it always works with 512-byte blocks.

5.25" Disks – Tracks and Sectors

The data on a normal 5.25" disk is stored in 35 circular tracks, numbered 0 through 34 ($00 through $22 in hexadecimal). The outermost track is track $00; the innermost track is track $22.

The disk drive, controlled by DOS, can position the read/write head (similar to the tape head in a cassette deck) under any one of the tracks. As the disk spins over the head, the drive can read or write the information on that track. The information is actually recorded on the underside of the disk.

Each circular track is divided (like a pie) into 16 sectors. The sectors on each track are numbered 0 through 15 ($00 through $0F). Each sector stores 256 bytes of usable data. DOS 3.3 always reads and writes information a sector at a time.

There are (35 tracks * 16 sectors =) 560 sectors on a standard formatted 5.25" disk. A disk can store a total of (560 sectors * 256 bytes per sector =) 143,360 bytes (140K).

Apple DOS 3.3, ProDOS 8 and ProDOS 16, Apple Pascal, CP/M, and Apple /// SOS all use the same track and sector formatting on 5.25" disks. However, the way the sectors are used for file storage varies greatly with each operating system.

3.5" Disks – Tracks, Sectors, Sides, Blocks

The data on a 3.5" disk is stored on 80 circular tracks. The tracks are numbered 0 to 79, or $00 to $4F in hexadecimal.

Most 3.5" disks are formatted as double-sided, which means they contain information on both the bottom and top surfaces of the disk media. Each of the 80 circular tracks is therefore divided into two "sides". The 3.5" drives have two read/write heads, one for the bottom surface and one for the top. The disk drive can position these heads — as one unit — under and over any track of the disk.

Each circular track (or side) is divided like a pie into several sectors. Each sector stores 512 bytes of usable data. The number of sectors per track is not constant for the whole disk, but varies depending on the track number.

Actually, each sector holds *524* bytes of data: the 512 data bytes are preceded by 12 "tag" bytes. On Apple II-format 3.5" disks, the 12 tag bytes are always zero, and are ignored during normal reads and writes. The 524-byte format is used for compatibility with the 3.5" disk format for Apple Macintosh computers, which store "file tag" information, related to file recovery, into the 12 bytes.

The outer tracks have a greater circumference (greater distance around the circle) than the inner tracks, and therefore have more media for storing information. The 3.5" disk format takes advantage of this for greater storage capacity, storing more sectors per track on the outer tracks:

Tracks	sectors/track	numbered
0-15 ($00-$0F)	12	0-11 ($0-$B)
16-31 ($10-$1F)	11	0-10 ($0-$A)
32-47 ($20-$2F)	10	0-9 ($0-$9)
48-63 ($30-$3F)	9	0-8 ($0-$8)
64-79 ($40-$4F)	8	0-7 ($0-$7)

For 3.5" disks, blocks and sectors are really the same thing; they're just numbered differently. A double-sided 3.5" disk contains a total of 1600 blocks or sectors. When thought of as sectors, they're numbered within each track and side as 0 to 11 (as shown in the table above). When thought of as blocks, they're instead numbered for the entire disk, 0 to 1599 ($0000 to $063F in hex). For example, block $0000 is the same as track $00, side 1, sector $0. Block $000C is the same as track $00, side 2, sector $0. Block $063F is the same as track $4F, side 2, sector $7 (the last block on the disk).

Because of the varying number of sectors per track, the conversion from block numbers to equivalent track, side, and sector numbers is a little less than straightforward. The Copy II Plus 3.5" Sector Editor will let you read any sector or block, and shows the numbers in both forms.

Disk Hardware – Reading & Writing Bytes, Speed

Most of this section deals with 5.25" drives, but 3.5" drives are very similar. The differences for 3.5" drives are discussed in the next section.

A 5.25" drive spins the disk at about 5 revolutions per second, or .2 seconds = 200 milliseconds per revolution.

The bytes on the disk (and the bits that make up those bytes) must be written at evenly spaced intervals around the circular track. Since the disk media is passing over the read/write head at a fairly constant speed, that means each bit must be written

onto the media at the right moment, in order to be placed onto the correct spot on the disk.

The timing involved in accessing the disk, especially when writing, must be precise. This makes disk access very "timing critical".

When writing a single byte to the disk, DOS sends the byte to a special "data latch" on the disk controller card. The hardware on the card then writes the 8 bits of the byte, one bit at a time, onto the disk media passing over the head. The hardware writes one bit every 4 usec (microseconds, or millionths of a second). It takes 32 microseconds to write all 8 bits of the byte (4 usec per bit * 8 bits per byte).

To write many bytes, DOS sends bytes to the data latch at exact 32 microsecond intervals, so that when the hardware has finished writing one byte, it receives the next byte to write.

If another byte isn't sent to the latch at the end of 32 microseconds, then the hardware begins writing individual zero bits onto the disk, a zero bit every 4 microseconds, until another byte is sent to the latch.

Any byte value can be written to the disk. However, only some values can be read back reliably, due to the Apple disk format and the nature of floppy disks in general.

When reading, the disk hardware waits until it reads a one bit from the disk, then gathers the next 7 bits to form an 8-bit byte. This is one of the fundamental limitations. Every byte read from the disk has its high bit set. If a byte is to be read back correctly, it must be written to the disk with its high bit set.

The other limitation is that the circuitry can't reliably read more than 2 zero bits in a row. If there are too many zeros in a row, the circuitry will begin reading some of them incorrectly as ones.

Bytes that have more than 2 consecutive zero bits are considered "invalid bytes", because they cannot be read reliably. If an invalid byte stored on the disk is read back, it might be read correctly, or it might be read incorrectly as another invalid byte or as a valid byte.

If a byte is read back as invalid, then some invalid byte is stored on the disk, though it may not be the byte that was read (since the circuitry may have read it wrong).

Since not all possible byte values can be read correctly, information being written to the disk must usually be "encoded" in some way first, so that only valid bytes are written. DOS does this encoding for every sector it writes.

Another problem in reading the disk is finding where one byte ends and the next byte begins. The data on the disk is stored simply as a long stream of bits. Here is an example bit stream:

```
1 1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1 0 0 1 0 1 1
```

The hardware could read a byte starting with any of the one bits. If the starting point is wrong, then the bytes read will be completely wrong. What is needed is a way to "synchronize" the hardware to the correct byte boundaries.

To synchronize the hardware to the bytes when reading, special bytes called "sync bytes" are written onto the disk with every sector. A sync byte is written by sending an $FF (binary 11111111) to the disk data latch, then waiting 40 microseconds before writing the next byte. The $FF is written during the first 32 microseconds, then the hardware writes 2 zeros to the disk before a new byte is sent to the latch. Sync bytes are sometimes referred to as 10-bit bytes (8 bits for the $FF + 2 zero bits).

If several sync bytes are written one after another, the following pattern will be stored on the disk:

```
1111111100111111110011111111001111111100 etc.
```

When reading this pattern from the disk, if the hardware is already "in sync", it will read 8 ones (to make an $FF byte), skip the 2 zeros (because it's waiting for another one bit), read the next 8 ones (to make another $FF), skip 2 more zeros, and so on. To DOS, sync $FF's look just like normal $FF's.

Often, however, the hardware will be "out of sync" when it begins reading the sync bytes. (For example, it may begin with the fifth one bit of the above pattern, and read back binary 11110011, or $F3.) Because of the 10-bit pattern being read 8 bits at a time, sync bytes have an interesting property. After reading at most 5 sync bytes, the hardware will always fall into sync with the bytes stored on the disk.

Other 9 and 10 bit patterns can also be used to synchronize the disk hardware, but 10-bit $FF's are the most common.

The total number of bits that can fit on a track is determined by how fast the disk is spinning when it is written to. If the disk is spinning at a slower than usual speed, then the bits will be written more closely together on the track. This means more bits are written before the track has completed a full revolution.

Unfortunately, the quality of the disk media imposes limits on how closely the bits can be packed reliably on the disk. The standard disk speed of 200 milliseconds per revolution was chosen as a good compromise between reliability and high data storage.

A standard disk speed also needs to be maintained for compatibility from one disk drive to the next. For example, a drive spinning at the slow speed of 210 milliseconds per revolution might be able to format, read, and write its own disks reliably, but it will have great difficulties reading a disk that was made on a drive that spins at a correct 200 milliseconds.

If a drive spins at the correct speed, 50000 bits will fit around the track. This can translate to 6520 ($1978) 32-usec bytes, or 5000 ($1388) 40-usec sync bytes.

Disk Hardware – 3.5" Drives

A 3.5" disk spins faster than a 5.25" disk. The rotational speed also varies: It spins more slowly on the outer tracks (about 150 milliseconds per revolution), to accommodate the greater number of sectors, and faster on the inner tracks (about 100 milliseconds per revolution) for the fewer sectors there.

The hardware writes a bit every 2 microseconds. It takes 16 microseconds to write all 8 bits of the byte. The timing for writing bytes isn't as critical as it is for 5.25" disks, because the Apple 3.5" circuitry provides its own timing for writing a new byte every 16 microseconds. The disk routines must simply be ready to give it the next byte "soon enough". If the disk routines don't deliver the next byte in time, the hardware leaves "write mode", not writing any more bytes to the disk until it has been reset.

Since the timing is fixed at 16 microseconds, or exactly 8 bit times, the disk routines can't use delays to directly write 10-bit sync bytes. However, it can write a repeating pattern of bytes which ends up on the disk as a series of sync bytes. The usual pattern is:

```
$FF      $3F      $CF      $F3      $FC      $FF
```

which is the bits:

```
11111111 00111111 11001111 11110011 11111100 11111111
```

Notice that this makes the usual bit pattern on disk (as described earlier) for sync bytes:

```
1111111100111111110011111111001111111100 etc.
```

Contents of a Sector

In order to read any given single sector, DOS must move the read/write head to the right track then begin reading bytes, waiting for that sector to pass by the head.

Every sector is made up of an address field and a data field. The address field contains information such as which sector this is and what the volume number of the disk is. The data field contains the actual information desired, such as a part of a file.

Here is a breakdown of a sector on a 5.25" disk:

Sync field: between 5 and 40 sync $FF's. This guarantees that the hardware is in sync when reading the sector.

Address Field:

Prologue: D5 AA 96. These 3 bytes acts as a marker, "A Sector Begins Here". The DOS read routines look for this pattern first. When it finds the pattern, it knows that the rest of the address field follows.

Volume number: 2 bytes. The volume number of the disk is stored next (in every sector) in an encoded form that uses only valid disk bytes. The encoding used here is called "4-and-4 encoding", and uses 2 bytes to store the 1-byte volume number. (A table of 4-and-4 encoded numbers is in Appendix E.)

Track number: 2 bytes. The track number is also stored in the address field of each sector, using 4-and-4 encoding. It is included so that in case the read/write head is "lost" and over the wrong track, DOS can find which track it's on by reading an address field, then move from there to the correct track.

Sector number: 2 bytes. The "hard" sector number, 4-and-4 encoded. (See below for hard and soft sectors.)

Checksum: 2 bytes. Another 4-and-4 encoded number that is used to verify that the volume, track, and sector numbers are correct.

Epilogue: DE AA. This marks the end of the address field.

Possible glitch bytes: See below.

Sync field: about 5 to 10 more sync $FF's.

Data Field:

Prologue: D5 AA AD. These three bytes mark the beginning of the data field. The encoded data always follows.

Data: 342 bytes. The 256 bytes of information are stored here, encoded as 342 valid disk bytes. The encoding scheme used is called "6-and-2 encoding", and involves some rather complicated bit rearranging, exclusive-ORing, and translation (through a lookup table) to valid disk bytes. The part of DOS that does the encoding and decoding is fast and efficient, but the 342 disk bytes bear little resemblance to the 256 data bytes they represent.

Checksum: 1 byte. This byte is used to help verify that there are no errors in the 342 data bytes.

Epilogue: DE AA. These bytes mark the end of the data field and the end of the sector.

Here is the breakdown of a sector on a 3.5" disk:

Sync field: between 5 and 100 sync $FF's. This guarantees that the hardware is in sync when reading the sector.

Address field:

Prologue: D5 AA 96. These 3 bytes act as a marker, "A Sector Begins Here". The 3.5" controller's read routines look for this pattern first. When it finds the pattern, it knows that the rest of the address field follows.

Track number: 1 byte. The track number is stored in the address field of each sector, translated to a valid disk byte. A table of the translated disk values is in Appendix E. (Actually, the track number, ranging from $00 to $4F is a 7-bit value, but the disk translation handles values only up to 6 bits. The highest bit of the track number is stored two bytes later.)

Sector number: 1 byte, as a translated valid disk value.

Side number and track high bit: 1 byte, translated. The side number for this side of the track and the high bit of the track number are combined together as a binary number in the form: n0000t, where n represents the side number (0 for side 1, 1 for side 2), and t represents the high bit of the track number. This 6-bit binary number is then translated to a valid disk byte.

Number of sides and interleave value: 1 byte, translated. The number of sides used on this disk and the "interleave value" are combined together as a binary number in the form: s0iiii, where s represents the number of sides (0 for single-sided, 1 for double-sided) and iiii represents the interleave value. (Interleave is described below.) This 6-bit binary number is then translated to a valid disk byte.

Checksum: 1 byte, translated, used to verify that the previous data values are all correct.

Epilogue: DE AA. This marks the end of the address field.

Possible glitch bytes: See below.

Sync field: about 5 to 10 more sync $FF's.

Data Field:

Prologue: D5 AA AD. These three bytes mark the beginning of the data field. The encoded data always follows.

Sector number: 1 byte, translated. The translated sector number is stored again here.

Data: 699 bytes. The 512 bytes of information (preceded by the 12 bytes of "tag" information) are stored here, encoded as 699 valid disk bytes. The encoding scheme is a variation of the 6-and-2 encoding scheme used on 5.25" disks; it involves a different method of bit rearranging (every 3 data bytes are converted to 4 consecutive disk bytes), 3 overlapping loops of exclusive-ORing, and translation to valid disk bytes. The 699 disk bytes bear little resemblance to the 524 data bytes they represent.

Checksum: 3 bytes. These three bytes, resulting from the 3 overlapping exclusive-OR loops, are used to help verify that there are no errors in the 699 data bytes.

Epilogue: DE AA. These bytes mark the end of the data field and the end of the sector.

Reading, Writing, and Formatting

A 3.5" drive controller (whether built into the computer or on a plug-in card) contains its own programming for handling reading and writing blocks and formatting the disk. ProDOS itself simply makes requests to the controller to perform the reading and writing as it needs them. Most other kinds of ProDOS-compatible disk devices, such as hard disks and RAMdisks, are the same way.

A 5.25" drive controller contains only enough programming to boot a disk; it does not provide for reading and writing sectors or formatting the disk. DOS 3.3 has its own set of internal routines (called RWTS, for Read Write Track Sector) for accomplishing this. ProDOS also has its own set of internal routines for reading and writing blocks (pairs of sectors) on 5.25" disks. It does not provide for formatting 5.25" disks, though. A separate application that includes its own formatting routines, such as Copy II Plus or Apple's Filer program, must be run to do this. Copy II Plus contains its own optimized 5.25" routines for formatting, reading, writing, verifying, etc.

When either reading or writing a sector, DOS (or whatever programming is doing the disk access) must first find the correct sector. It calls a read address field routine that looks for and reads the next address field to pass by the read/write head. DOS then checks the track and sector numbers from this address field to see if this is the desired sector. If it is not, DOS continues to look for the correct one. If it can't find the desired sector within a certain number of tries, it gives up and returns an error.

When reading, after DOS finds the right address field, it calls a routine to read the data field, which will be passing by the read/write head within a couple hundred microseconds.

When writing, after finding the correct address field, DOS calls a routine to write a new data field over the old one. The calls themselves aren't timed exactly, so DOS might begin writing the new data field a few bits earlier or later than the old data field. This produces a "glitch" on the disk where writing begins, since the new bits aren't in sync with the previous bits on the disk.

Another glitch occurs at the end of the data field, when DOS stops writing new information.

When DOS reads the disk, these glitches often throw the hardware out of sync with the bytes on the disk. That's why

both address and data fields are preceded with sync fields, so that the hardware can get back into sync.

Notice that during normal use, data fields are rewritten, but not address fields. When a disk is formatted, both address and data fields are written onto the disk.

In formatting each track, DOS writes a very large initial sync field, then all of the sectors for that track in one revolution of the disk. This "wipes clean" any old information that might have been on the track. The data fields written are "empty". (When read and decoded, the sectors contain all zero bytes.)

The initial sync field is large enough that the last sector put onto the track will overwrite the beginning of the sync field as the disk completes one full revolution.

If the disk is spinning too fast (5.25" disks only), then the entire initial sync field (and possibly part of the first sector) will be overwritten, which means the formatting failed. If the disk is spinning more slowly than usual, then the remaining part of the sync field which was not overwritten will be very large.

When DOS 3.3 begins formatting a 5.25" disk, it writes and rereads the first track a few times, adjusting the sizes of the sync fields between each sector (changing the amount of data written onto the track) so that the remaining initial sync field is about the same size as as the other sync fields. This certainly isn't necessary, but it spaces the sectors around the track a little more evenly.

A 3.5" controller double-checks and adjusts the drive speed during formatting, so the remaining part of the initial sync field is about the same size on all 3.5" disks.

Hard and Soft Sectors (5.25" disks)

Before writing a sector, DOS must "pre-nibblize" the 256 data bytes into 342 disk bytes to be written. After reading a sector, DOS must "post-nibblize" the 342 disk bytes back into 256 data bytes. Because of the time this takes, the next sector to read or write has already passed by before DOS is ready to access it. DOS is fast enough, though, to access every other sector as it passes under the head.

To make disk access fast yet simple, DOS 3.3 "re-maps" the sector numbers in memory so that if a program asks for consecutive sector numbers, DOS will actually access every other disk sector for speed. The sector numbers asked for by a program (including the Copy II Plus Utilities) are called "soft sectors". The sector numbers actually stored on the disk are called "hard sectors". For example, if you access soft sectors $7, $6, $5, and $4 in that order, DOS 3.3 will look on the disk for hard sectors $1, $3, $5, and $7.

For ProDOS, 8 blocks fit on each 16-sector track. The blocks are numbered 0 - 7 on track 0, 8 - 15 on track 1, 16 - 23 on track 2, etc. Similar to DOS 3.3, the numbering between blocks and hard sectors is done in a special way to keep disk access fast.

Here is a table that shows the hard sector numbers for a track, the corresponding DOS 3.3 soft sector numbers, and the ProDOS block numbers (divided into part 'a' and part 'b' since each block is 2 sectors long.):

Hard Sector	DOS 3.3 Soft Sector	ProDOS Block Number
0	0	0-a
1	7	4-a
2	E	0-b
3	6	4-b
4	D	1-a
5	5	5-a
6	C	1-b
7	4	5-b
8	B	2-a
9	3	6-a
A	A	2-b
B	2	6-b
C	9	3-a
D	1	7-a
E	8	3-b
F	F	7-b

To translate the hard sector number into the actual 4–and–4 encoded sector number stored in the address field, see Appendix E.

Interleave (3.5" disks)

Before writing a block, the 3.5" controller must "pre–nibblize" the 512 data bytes (plus the 12 "zero" tag bytes) into 699 disk bytes to be written. After reading a block, the controller must "post–nibblize" the 699 disk bytes back into 512 data bytes. The pre–nibblize and post–nibblize routines take different amounts of time on different types of drives and controllers.

A Unidisk 3.5 drive, for example, actually contains an internal "computer" which performs the pre– and post–nibblizing within the drive, transferring the finished 512 bytes of data to and from the Apple. Because of the time this takes, three additional sectors have passed by before the internal controller is ready to access another sector.

For an Apple 3.5 drive connected to an Apple IIGS, the 3.5" controller inside the GS performs the pre- and post-nibblizing. One additional sector passes by during this time.

Because the controller isn't fast enough to access consecutive sectors on the track, Apple-format 3.5" disks are formatted with the sector numbers out of order, or "interleaved", on each track. Here is the order of the sectors for the different track groups:

Track	Sectors/track	Sector numbers in order (hex)
0-15	($00-$0F)	12	0 3 6 9 1 4 7 A 2 5 8 B
16-31	($10-$1F)	11	0 3 6 9 1 4 7 A 2 5 8
32-47	($20-$2F)	10	0 5 3 8 1 6 4 9 2 7
48-63	($30-$3F)	9	0 7 5 3 1 8 6 4 2
64-79	($40-$4F)	8	0 2 4 6 1 3 5 7

Notice that with this order, consecutive sector numbers are always four sectors apart, so the disk has an "interleave" of 4. This allows an application to request consecutive block or sector numbers, accessing the blocks as fast as a Unidisk 3.5 drive allows.

This interleave is formatted into the "hard" sectors on the 3.5" disk (the sector numbers are actually out of order), as opposed to the "soft sector" renumbering that occurs with 5.25" disks (the hard sector numbers on disk are in order, but a lookup is done from the "soft" sectors that a program requests).

Macintosh disks are formatted (by block) the same way as Apple II-format 3.5" disks, with two exceptions: (1) The 12 tag bytes within each Macintosh block normally contain information related to file recovery, rather than containing zeros. (2) Macintosh disks are formatted with an interleave of 2 (rather than 4), which means consecutive sector numbers are two sectors apart around the track.

You could copy a Macintosh disk using the Utilities COPY DISK W/ FORMAT option. However, the Macintosh tag information would be lost, and the interleave of the resulting disk

would be wrong. This would cause disk access to the copy to be slower than the original disk, because the sectors aren't in the proper order for fast, efficient access. Similarly, you could copy an Apple II–format 3.5" disk using the SECTOR COPY option found in CPS's Copy II, Apple Macintosh version, but again the interleave would be wrong, causing disk access to the copy to be slower.

The SECTOR COPY option in the Copy II Plus 3.5" Bit Copy program, however, correctly copies any tag information, and also preserves the interleave of the original disk on the copy. Both Macintosh disks and Apple II 3.5" disks can be copied with Bit Copy 3.5.

(Note, however, that the Macintosh "HFS" file system, the way the Macintosh uses the blocks, is very different from Apple II ProDOS. The Copy II Plus Utilities and other ProDOS utilities are not designed to catalog or work with Macintosh files.)

Appendix B – Disk Protection Schemes

Protection?

What makes a disk "protected"?

In Appendix A, the format of a normal DOS 3.3 or ProDOS sector was given. Standard disk copy programs look for this format on every track of the disk. If the prologs and epilogs can be found in the right places and the checksums match with the data, then the Disk Operating System can be "confident" that the data itself is correct. This helps to produce a very reliable copy.

The simplest protection schemes simply change this format slightly. Since a normal DOS then can't find the byte patterns it's looking for, it doesn't know how to make sense of the disk data. It gives up and prints an enlightening message such as "I/O ERROR". In other words, any change from a standard disk format, if it was put there to make copying more difficult, can be considered a "protection scheme". The sophistication of the changes varies greatly. Many protected formats bear no resemblance to standard sectors at all.

There are two possible approaches to copy protection. The first is to store the program information on the disk in such a way that a bit copier can't reproduce all of it. When you try to boot the copy, the program is incomplete and won't run. The second approach is to store the program in a reasonably normal form, but also put special bytes or patterns which are difficult to copy somewhere on the disk. When you boot this, the program loads correctly, but then promptly begins by checking that the special bytes are still on the disk. If they are missing or incorrect, then the program "knows" that this is a copy, and will refuse to run.

Perfection?

Why can't a bit copy program just copy "everything" on the disk?

There are a few reasons for this. The most pervasive one has to do with the fact that on a circular track, there is no defined "beginning" or "end". A bit copy program must begin reading at some arbitrary point around the track, and then make sense of what it reads. After reading two or three revolutions of the track into a memory buffer, the bit copy program can find any given byte from the track two or three times in the buffer. The number of bytes between these identical images is how many bytes were on the original track.

If all drives spun at exactly the same speed, then the bit copier could, starting at any byte, write the correct number of track bytes onto the duplicate disk. These bytes would exactly fill the circular track on the duplicate disk. The last data byte written would fall just before the first one on the track. But if the duplicate drive spun too fast, then the end of the track image would overwrite the beginning, destroying part of the data. If the drive spun too slowly, then there would be a gap between the beginning and the end. This is unacceptable, since the gap or the overlap could end up in the middle of a data area. Disk drive speed varies too much (even on a given drive) to copy a track this way.

Most disks are written with first a large sync field, then the data area. The end of the data area overlaps part of, but not all of, the sync field as the disk completes a full revolution. (See Appendix A.) The size of the remaining sync field is determined by how fast the drive that made the disk was spinning. If a bit copy program can identify the beginning and ending of the data area, it can also write a large sync field before the data area. The resulting sync field may be a slightly different size than on the original, but in most cases that doesn't matter.

Therefore, one of the tasks for a bit copy program is to identify the start and end of the "useful" data area on each track. Then when it writes the track, it can let the "sloppiness" caused by varying drive speeds fall outside of this data area, where it can hopefully be ignored. Many protection schemes involve making it difficult for a bit copy program to find the start and end of the track data.

The first protection schemes involved very simple changes, since there weren't any programs yet available that could copy these disks. When bit copy programs that could back up these disks were developed, more complicated protection schemes were invented. New copy programs were released to copy the new protection schemes, and new schemes were created to "beat" the bit copy programs. This cycle still continues. The following descriptions start off with the easier changes and progress to some of the state of the art schemes currently in use.

Changed Address and Data Headers

As mentioned earlier, standard disk copy programs expect to find normal sectors on the disk, with correct prologs, epilogs, checksums, etc. These header values can also provide clues to a bit copy program to help it find the track start and end, since it knows that a sync field usually precedes every D5 AA 96 address prolog.

Since Apple DOS looks for these bytes when reading a sector, changing these to new values (e.g. D5 AA 97) will cause any normal copy program to fail. Prologs, epilogs, track numbers, and checksums have all been changed in various schemes. This was one of the first and most simple disk protection schemes developed, but even today most disks employ this as one of their protection methods.

Changed Sync Bytes

The first bit copy programs didn't look for address prologs at all. Instead, they looked for the large sync $FF fields and determined that a track started right after one of these. Soon, many copy-protected disks used both changed address headers and changed sync fields. One of the most popular changes was to write sync $FE's rather than $FF's. The bit copiers responded by being able to recognize a range of values as sync bytes, including both $FE and $FF.

Some disks instead had large gaps of invalid bytes (bytes with more than two consecutive zeros), followed by only the minimum number of sync bytes required by the hardware. Without familiar headers or large sync fields, the bit copy programs had nothing to use to reliably determine the start of a track. However, the invalid bytes couldn't be important data areas, since they can't even be read reliably, and so were probably part of the track-end gap. With this knowledge, new bit copy programs were written that included subroutines to convert invalid bytes to some known value, usually sync bytes.

About this time, the concept of parameters was introduced to bit copy programs. It became obvious that no single set of algorithms would be able to automatically handle all types of copy protection. The user needed to be able to turn certain routines on or off, and to set the operating values for others.

Synchronized Tracks

Somebody got clever one rainy day and decided the real way to copy-protect a disk was to change nothing that is visible on the track, just change the alignment of the information from track to track. When DOS formats a disk, the tracks are always written with a certain circular alignment, due to the timing consistency of the formatting routine. As an example of this alignment, suppose a program reads sector 0 from track 0, then immediately

steps to track 5 (which always takes the same amount of time), then begins looking for a sector. The first sector to pass under the head will always be sector $C, because sector $C just happens to lie in the right place for this to happen.

Most copy programs and formatting programs all produce different alignments, because they spend varying amounts of time on each track before stepping to the next track. This usually doesn't hurt anything. However, a copy-protected disk can be created with a certain fixed alignment, then this alignment can be checked by the protected program whenever the disk is booted. If the alignment differs, then the program "knows" this is a copy and not the original disk, so it refuses to run.

Bit copy programs began including an option to handle synchronized tracks (not to be confused with "sync bytes"). They copy not only the data, but whatever track alignment is on the original disk as well.

Half Tracks

This method appeared about the same times as synchronized tracks. The Apple 5.25" disk drive can actually position to 70 different tracks, not 35. Unfortunately, the read/write head used in the drive is too wide to write complete tracks on every track boundary. It would overwrite the information stored on adjacent tracks. So DOS actually steps the head twice for every track on the disk, giving the familiar 35 tracks. But since it is possible to position the head to any of the 70 half-tracks, some disks shift the data and start using tracks on half-track boundaries. For example, rather than writing information on tracks 0, 1, 2, 3, etc., they might use 0, 1.5, 2.5, 3.5, etc. Any possible pattern can be used, as long as the increment is at least one whole track.

There is no easy, foolproof way to determine what half-tracks are used by a protected 5.25" disk. In general, if you try to read

(with the nibble editor) a track or half-track that was never written to, you will see large areas of invalid bytes. If data was written to the half-tracks on either side, you may see a few areas that look like valid track data, as the wide read head occasionally picks up these bytes from either side. The HI-RES DISK SCAN option can help you find the half-tracks containing valid data. (Try using HI-RES DISK SCAN on a normal 5.25" disk, setting the track increment to .5 to see the invalid half-tracks along with the valid tracks.)

Copy II Plus can position the drive head over any half-track, or even quarter-track! To do quarter-tracking, the Bit Copy program instructs the drive to begin stepping from one half-track to the next, then it stops the positioning while the read/write head is still moving. The head is left positioned about halfway between the two half-tracks.

An Extra Track?

The hardware can (on most drives) write one extra track after the last track on the disk. This would be track $23. Since a normal copy program doesn't suspect that an extra track exists, it won't try to copy it. This is part of the reason bit copy programs such as Copy II Plus allow you to specify start and end tracks to copy.

Bit Insertion

Remember that sync bytes are bytes written with extra zero bits on the end. Groups of sync $FF's are written to ensure that the hardware will synchronize to the data on the disk. Well, nothing prevents you from putting an extra bit on the end of other bytes, as long as the maximum number of consecutive zeros is not exceeded. Whenever the program must access the disk frequently (for reading data files or other information), this scheme is easy, since it doesn't interfere with any DOS routines. This is why so much business software uses it.

Whenever one of these programs is booted, it finds the spot on the disk where it knows these special "bit-inserted bytes" should be. It then uses a carefully timed routine to determine if the extra bits are there. (See Appendix A for the timing between bits and bytes.) If not, it knows this is a copy, and refuses to run.

Earlier bit copy programs could not determine which bytes on the disk were sync (9 or 10 bit) bytes. The timing involved in reading and storing each byte into memory and checking for sync at the same time makes this very difficult. The early copy programs instead made "educated guesses" as to where the sync bytes were. The more recent versions of Copy II Plus use a more sophisticated read routine and can determine sync with a fairly high degree of reliability. These bytes appear in the nibble editor as inverse.

Nibble Counting

You can adjust the speed of your Apple 5.25" disk drives. They normally run at about 300 rpm (200 milliseconds per revolution), but this can vary significantly, even on a single drive. As mentioned earlier, this affects the number of bytes that will fit on a track. Some software publishers take advantage of this fact. When making a commercial disk, the duplication program will write a track, then re-read it to find out out many bytes (or nibbles – both terms are used) are on the track. It then writes this count on the disk somewhere. When the disk boots, this count is compared to the actual number of bytes on the track and if they are equal (or within a specified tolerance), the program will run. However, even very small speed variations will affect the number of bytes on a track, so it is unlikely that your drives will produce the exact same count as the drive which was used to produce the original disk.

Bit copy programs respond by varying the nibble count somewhat without adjusting the drive speed. (The method used is

explained in Appendix C.) Note that the nibble count naturally comes closest if the speed of the duplicate drive closely matches the speed of the drive that the disk was originally made on. The speed of your original and duplicate drives do not have to exactly match each other to do accurate nibble counting. (Remember that adjusting drive speed applies only to 5.25" drives; 3.5" drives dynamically adjust their rotational speed from one track range to another.)

Long Tracks

Some 5.25" protected programs are written with a large amount of data on each track. The drives that make these disks are slowed down slightly so that the extra data will fit. If you try to copy the disk with a normal-speed duplicate drive, the end of the long track will overwrite the beginning, creating an unbootable disk. This is one possible cause of an error 5 (write verify error) when backing up a disk with Copy II Plus.

When this protection scheme is used, the best solution is to simply adjust your 5.25" drive to a slightly slower speed so that the track will fit on the duplicate disk. Unfortunately, if you leave your drive at a slower speed, it may be slightly less reliable when accessing "normal-speed" disks (disks that were made on a drive that spins at the correct speed). If you have two drives, here is a compromise suggestion: Set drive 1 to spin at 200 milliseconds per revolution for greatest reliability. Then set drive 2 to spin at a slower 200.5 to 201.0 milliseconds, which will help back up protected disks while still maintaining good reliability.

Write-Protect Check

When you use a disk that has a tab over the write-protect notch, this does two things. The electronics in the drive prevent any program from writing to the disk, and a "flag" is set which the program can check to see if the disk is write-protected. Some

commercial disks have no notch, and so are permanently write-protected.

Some protected programs (that have no notch in the disk) check the write-protect flag when they are booted. If the flag says "not write-protected", then the program knows that this is an ordinary notched disk, and must be a copy rather than the original disk. It will then hang, or reboot, or ask you to insert the original. (It could also trash the data on your backup.) If you put a write-protect tab over the backup before you boot it, then the program cannot use this to determine that a copy is running.

There is no ready way to determine when this protection scheme is being used. If you want to be on the safe side, if the original disk is write-protected, always put a write-protect tab on your duplicate disk before you boot it. If the original is not write-protected, don't put a tab on the backup.

"Non-sync Sync"

A few protected programs use a pattern of normal 8-bit bytes to synchronize the hardware to the disk data. This pattern usually has to be fairly long and consist of the proper bytes in order to synchronize correctly. If this scheme is used, then 9 and 10 bit sync bytes are not needed, making it more difficult for bit copy programs to determine the track start and end.

This covers the main schemes currently in use. It should be noted that several disks use combinations of the above schemes just to make things more complicated: radically different sector formats, with different headers on different tracks, short sync fields or almost no sync at all, half-tracks, etc. ad infinitum.... In some cases, the combinations form almost a new protection scheme in itself. Here is one example:

Spiral Tracks

This method combines synchronized tracks with half-tracks to store data in an unexpected way. Remember from the discussion of half-tracks that the 5.25" Apple disk read/write head is too wide to write complete tracks on every half-track boundary. But this doesn't prevent it from writing a smaller amount of information on each half-track (just a portion of the circular track), as long as it won't interfere with the data on adjacent half-tracks. A disk with spiral tracks is created by writing about 1/4 the normal amount of information stored on a track, then stepping to the next half-track and doing the same. This process is repeated until all the information is written to the diskette. Since each track portion is short, it never overwrites or interferes with the track portion on the half-track before or after it. If you try to copy this disk without synchronizing, the half-track images will overwrite each other, and the copy will not work. Copying is made even trickier because the read/write head on the original drive may pick up some information from the adjacent half-tracks, making it harder to find the track start and end.

One technique that helps to copy a disk that uses spiral tracks is to read and write on quarter-tracks, between two half-tracks. The drive can read the two track arcs on either side in one revolution of the disk.

Appendix C – Routines and Parameters

This appendix describes the methods Copy II Plus uses to copy a disk, and how the various parameters affect the copy process. Each parameter has both a number and a name. The name provides a quick way to remember what each parameter does. If a parameter represents a disk byte value, it can be stored normally (for example, $FF) to represent a normal 8-bit byte, or with its high bit clear ($7F) to represent a sync byte. If the byte is part of a byte pattern to search for in the buffer, a zero value in the parameter means "match anything for this byte".

Bit copying is more complicated than sector copying, and it is explained first.

When bit copying, Copy II Plus begins with the READ A TRACK routine. This simply reads bytes from the original drive until it fills the buffer. Copy II Plus uses one of two possible read routines. It normally uses the routine that checks if each byte is a sync (9 or 10 bit) byte as it reads it. However, if you change parameter 56 (OLD.READ) from 0 to 1, Copy II Plus will use the old read routine which reads everything as nonsync (8 bit) bytes.

Every byte read by the drive has its high bit set. If it is a normal 8 bit byte, Copy II Plus stores it in memory as it was read, with its high bit set. If it is a sync byte, Copy II Plus clears the high bit (subtracts $80 from the number), and stores this new value in memory. When the track buffer is displayed, all numbers with their high bit clear are displayed in inverse with the high bit set again. For example, a sync $FF from the track is stored in memory as a $7F, and is displayed on the screen as an inverse $FF. This information is helpful when setting some of parameters discussed below.

If parameter 9 (CLEAN?) has been changed from 0 to 1, then the CLEAN SYNC FIELDS routine is called next. This routine looks for the areas between the end of each data field and the

beginning of the next address field, and between the end of each address field and the beginning of the following data field, and sets all bytes within these areas to standardized sync (usually sync $FF's; the actual value is stored in STAND, parameter 7). To find the end of the address or data field, it usually looks for the epilog bytes DE AA XX, but these values are from parameters 19, 1A, and 1B (ADDRESS.END) and can be changed. To find address or data start, it matches the first two bytes from either ADDRESS.START (parameters E and F) or DATA.START (parameters 1C and 1D), which usually contain D5 AA.

If parameter 31 (FIX.INVALID?) is changed from 0 to 1, then Copy II Plus next calls the FIX INVALID BYTES routine. This routine scans the buffer for occurrences of invalid bytes. There are bytes that the hardware cannot read reliably (those with more than two consecutive zero bits). It will replace any invalid bytes with standardized sync bytes (from STAND, parameter 7). These are the bytes it will convert to standard sync:

```
81 82 83 84 85 86 87 88 89 8A
8B 8C 8D 8E 8F 90 91 98 A0 A1
A2 A3 B0 B1 B8 C0 C1 C2 C3 C4
C5 C6 C7 C8 D0 D1 D8 E0 E1 E2
E3 E8 F0 F1 F8
```

In addition, Copy II Plus always looks for $80's in the track buffer and changes them to standard sync, whether or not parameter 31 is set to 1.

It then calls the STANDARDIZE SYNC routine, if parameter 8 (STANDF) has been changed from 0 to 1. This routine looks for nonstandard sync fields and changes them to standard sync. It is good for cleaning up sync fields that contain a mixture of sync bytes, and a few other "stray" values.

It looks for fields of at least SYNC.# (parameter 6) bytes that have been marked as sync by the read routine. The field can contain up to GLITCH.SIZ (parameter 32) consecutive bytes that are not

sync. The bytes are then all converted to standard sync, the value contained in parameter 7, STAND. If CHANGE (parameter 33) is 1, the glitch bytes are also changed; if CHANGE is 0, they're left alone.

The next task of Copy II Plus is to find the start and end of the track data. There are two different methods it can use to determine the track start. The methods it uses are controlled by parameter 55, FIND.START. If this is set to 3, Copy II Plus will try first by "header". If this fails, then it will try by "sync". If set to 1, it will try first by "sync" then header. When it finds the track start, it will display either "HEADER" or "SYNC" in the center window to show you which method it used.

The FIND HEADER routine looks for an address header (part of or all of the address field) to determine the track start. It tries to find the pattern of bytes from ADDRESS.START up through ADDRESS.END (parameters E to 1B) in the track buffer. If it can match the first MATCH bytes (parameter A), then this is the track start. The ADDRESS.START table contains 3 bytes for the address prolog, and 8 bytes for the encoded volume, track, sector, and checksum. ADDRESS.END immediately follows and contains the address epilog bytes. A zero byte in any of these parameters will match any value from the track buffer. The FIND HEADER routine often requires several parameter changes before it can find the track start, since many protected disks use changed headers. If no match is found, this routine "fails", and the FIND SYNC routine is tried.

The FIND SYNC routine will attempt to find the track start by looking for the largest group of valid sync bytes in the first part of the track buffer. The sync field must be at least SYNC.# (parameter 6) bytes long. It can contain small glitches of non-sync or invalid bytes. The track start is set to the end of this field. Since most disks have a large sync field before the track start, this routine will correctly find the track start most of the time. If no valid sync fields can be found, this routine "fails", and the FIND GAP routine is tried.

Copy II Plus then must determine the end of the track data. It looks for a duplicate image of the track start later in the buffer, then moves back over the last sync field or other garbage that may be present. You can also instead have it set the track end as a fixed number of bytes after the track start. Lastly, you can have it set the track end to a fixed number of bytes after the last data prolog.

The standard track end finder first skips TRKMIN (parameter 3) pages past the track start. It then starts looking for at least EMATCH (parameter 50) bytes that match the track start. This is the repeat image of the track start later in the buffer. It then backs up over any sync field or other garbage that may be at the end of the track. The sync field can contain up to GLITCH.SIZ (parameter 32) consecutive non-sync "glitch" bytes. This point is the track end.

If you want to instead set the track end by cutting the track off a certain number of bytes from the track start, change parameter 44 (CUT?) from 0 to 1. The number of bytes to cut from (the desired track length) should be stored as a two-byte number in CUT.HIGH (parameter 45) and CUT.LOW (parameter 46).

(Note: This parameter exists only in the 5.25 inch bitcopy.) If you want to set the track end a fixed number of bytes after the last data prolog (usually D5 AA AD), then set parameter B3 (ALTEND?) to 1. This routine is helpful when the big gap is full of garbage and you want to set the track end precisely at the right spot. Set the pattern to look for in parameters B4-BC (ENDPAT). Set the length of this pattern in parameter BD (ALTMAT). Lastly, put the number to add to the matched pattern to set the track end in parameters 45 (CUT.HIGH) and 46 (CUT.LOW). ENDPAT defaults to the standard data prolog (D5 AA AD), and ALTMAT is set to the length of this pattern, 3. All you have to change is the length to add (CUT.LOW and CUT.HIGH) and ALTEND? to enable this routine.

If parameter 4F (SDFLTR) is changed from 0 to 1, Copy II Plus adds an extra check as it analyzes the data for track start and end. SDFLTR stands for Single Density FiLTeR. This check verifies that the data between track start and track end does not contain more than 1 consecutive zero in each byte. If it does, Copy II Plus continues to look for another track start and end. This check is most useful when copying disks that use track arcing and contain 4-and-4 encoded data; it helps keep spurious data in adjacent half-tracks from confusing the Bit Copy program.

If the track data is more than TRKMAX (parameter 2) pages long, it assumes the analyze routines failed. If Copy II Plus cannot find the track start using any of the methods selected by parameter 55, it re-reads and re-analyzes the track up to EREAD (parameter 0) times. If it still cannot find the track start, then a READ ERROR occurs. An error number 2 appears in the status display, and Copy II Plus simply grabs a block of data from the buffer that would be about the correct length for a normal disk, and uses this for track start and end.

If parameter 34 (BIT.FLAG) has been changed from 0 to 1, then the BIT INSERT routine is called next. This routine scans through the track data looking for a pattern of up to 5 bytes. If this pattern is found, the matching bytes in the buffer can be changed to either sync or non-sync bytes. This routine can be used when the protected program is checking that a certain byte on the track is a sync byte. However, note that in nearly all cases, Copy II Plus will correctly identify all sync bytes automatically as it reads the track, so the BIT INSERT routine is not needed often. The exception is when 9-bit sync bytes are used, as the read routine can not always correctly distinguish these from 9-bit bytes due to the tight timing requirements and limited flexibility of the 65C02 processor.

The 5 bytes that BIT INSERT tries to match are stored in the BIT.TABLE, parameters 35 through 39. The pattern matching ignores the high bits of each byte. The values in the table can

have their high bits either cleared to 0 or set to 1. This indicates whether the bytes should be written as sync or normal bytes. When a match is found, the corresponding high bits in the track buffer are also set or cleared, will will cause the write routine to write them as normal (8 bit) or sync (9 or 10 bit) bytes. Any zero values in the BIT.TABLE will match anything.

Copy II Plus then calls the WRITE TRACK routine to write the track data in memory to the duplicate disk. It starts writing from a few bytes before the track start to include the preceding sync field (if there is one), and continues to the track end. It writes all sync as either 9 or 10 bit bytes, depending on the value of BITS, parameter 3E. If BITS is set to 1, 9 bit bytes will be written; if set to 2, 10 bit bytes are selected. If the value of parameter 4D, ERASE, is 1, then the entire track is erased to sync $FF's before the track data is written. If ERASE is changed to 0, or if the track increment is less than one, then the track is not erased first, and only 16 bytes are written before the selected track start (this is to prevent overwriting adjacent half-tracks).

If parameter 47 (VERIFY?) is left at one, it then immediately calls WRITE VERIFY to verify that the track just written is correct. (If changed to 0, it will not verify.) This routine simply checks that the track start was not overwritten by the track end (track too long). If this test fails, Copy II Plus first calls the TRACK CHOPPING routine. This chops a track that is too long by shortening all the sync fields to a length specified in KEEP (parameter 3D). The chopped track is rewritten and verified again. If the verify still fails after EWRITE (parameter 2) retries, a write verify error (error 5) appears in the status display. WRITE VERIFY also fails if there is no disk in the duplicate drive.

If you've answered Yes to the KEEP TRACK LENGTH question, or changed parameter 4B (DONIB?) from 0 to 1, Copy II Plus next calls the NIBBLE COUNTING routine. This routine computes the number of bytes (nibbles) on the original disk and tries to maintain that count on the duplicate disk. It works by converting some of the normal bytes to 9 or 10 bit bytes if there

are too many bytes on the duplicate disk, or by converting sync bytes to 8 bit bytes if there are not enough. (This works on the principle that by adding bits to some bytes, the bytes take up more space on the duplicate track, so fewer of them are needed to fill the track.) It calculates the number of bytes to convert based on the current setting of BITS (9 or 10 bit sync?), and the difference between the length of the original track and the length of the duplicate track. The difference is compared to TOLERANCE (parameter 4C) and if it is less than or equal to this number, the nibble count succeeds. Otherwise, it compares again and rewrites the duplicate track. It may take several tries before the nibble count matches. If there is more adjustment to do but no more bytes which can be changed, a nibble count error (error 6) is displayed for this track.

If you've answered Yes to the SYNCHRONIZE TRACKS question or changed parameter D (DOSYNC) from 0 to 1, Copy II Plus also maintains SYNCHRONIZED TRACKS as it copies. This routine makes sure that the information on the duplicate disk has the same track-to-track alignment as on the original disk. SYNC.TRACK (parameter C) is the reference track to synchronize with (usually track 0). SYNC.START (parameters 22 through 2F) is a table of bytes to match to find the start of the reference track. It currently contains the address field bytes for sector 0. SYNC.MATCH (parameter 30) is the number of bytes in the table to match. If the SYNC.START bytes cannot be found on the reference track, Copy II Plus will spin the disk indefinitely looking for them. This will only happen if you're trying to synchronize on a nonbootable disk. Press [RESET] to recover.

If parameter 51 (DYNAM) is changed from 0 to 1, the DYNAMIC HEADER CHANGE routine is also used. Some disks change the address header for each track on the disk. They usually store the new header at the end of the current track. Using this routine, you can tell the Bit Copy program where to find the new header and it will dynamically update the address header table.

The new header is found by adding the offset in parameters 52 and 53 (DYNAM.LOW and DYNAM.HIGH) to the start of the track. Parameter A (MATCH) is used to determine the length of the header (number of bytes to fill into the header table). Parameter 54 (FILL.ORDER) determines whether to fill the header table forwards (0) or backwards (1).

Sector copying is more straightforward than bit copying. The sectors from each track are read from the original disk, then formatted and written onto the duplicate disk. Without any parameter changes, normal DOS 3.3 and 3.2 disks can be copied reliably. By changing a few parameters, many protected disks can also be copied.

The parameters used in sector copying are very similar to the custom patch values that are used in the Sector Editor. A good knowledge of address and data field formats helps in understanding these parameters.

When reading, Copy II Plus looks for address prologue bytes that match APRO, parameters 57 through 59. The seed value to use when calculating the address field checksum is in parameter 5A, ASEED. Address checksum errors are detected if parameter 5B, ACHKF, is nonzero. The first two address epilogue bytes are checked against AEPI (parameters 5C and 5E) if AEPIF (parameter 60) is nonzero.

The three data prologue bytes must match DPRO, parameters 61 through 63. The data checksum seed value is stored in parameter 64, DSEED. The data field checksum is tested if DCHKF, parameter 65, is nonzero. The first two data epilogue bytes must match DEPI (parameters 66 and 67) if DEPIF (parameter 6B) is nonzero.

If DOSFLG, parameter 77, is zero, then the sector copier will automatically try to copy using DOS 3.2 format first. If this fails, then it tries copying using DOS 3.3 (same as ProDOS) format. If DOSFLG is nonzero, it tries only DOS 3.3 format.

When writing, the three APRO bytes are used for the address prologue. The seed value in ASEED is used to determine the address checksum. If AEPIF is nonzero, then the 4 epilogue bytes from AEPI (parameters 5C through 5F) are written. If AEPIF is zero, then the address epilogue bytes read from the original disk are used instead.

The three data prologue bytes are used from DPRO. DSEED is used as a starting seed value in writing the data field and checksum. If DEPIF is nonzero, the 5 epilogue bytes from DEPI (parameters 66 through 6A) are used. If DEPIF is zero, then the data epilogue bytes read from the original disk are used instead.

During writing, if parameter 76, FNYFLG, is nonzero, then 5 "funny" sync bytes are written before each address field. These bytes help copy some protected disks, including the older PFS series disks. Rather than writing the last 5 sync $FF's, the five bytes from FUNNY (parameters 6C through 70) are written. The number of extra zeros to add to each funny bytes are stored in TIME, parameters 71 through 75.

For the 3.5" Bit Copy, parameter 64 is instead called TRKFLG. It left at 0, it will allow incorrect track and side numbers in the address field of a sector. If set to FF, it requires that the track and side values be correct. Parameters 6C, 6D, and 6E (DTSEED1, DTSEED2, DTSEED3) are the three checksum seed values for the data field.

Special 5.25 inch Bit Copy Parameters

The following are special routines that have been developed to handle disks that otherwise could not be copied for backup. These routines only exist in the 5.25 inch bitcopy. The 3.5 inch bitcopy will develop its own set of special routines as needed which will be documented as they evolve. Please note that these

parameters are very technical in nature and require a high degree of experience to be used properly.

PARAMETER 78 (LSPARM) - SECTOR COPY ONLY

This routine was originally developed to handle a protection scheme that remarkably enough first appeared on Locksmith 5.0, hence the name "LSPARM". It has since appeared on numerous disks and is perhaps one of the most widely used protection schemes. This parameter also will set an automatic check for "Prolok" protected disks, a scheme that burns a laser hole into the disk and checks for the bad data.

Here's how this parm works:

When set to 1, LSPARM will write a special sector specified by parms A1 (track number) and A2 (sector number). This sector is an empty sector filled with some E7 bytes where the number of zero bits written after each E7 varies (some have none, others have 1, still others have 2). When the protection scheme checks the disk, it "knocks the data out of sync" so it can see these extra zeros as data bits. If the right number of zeros is not appended to each byte, then the disk is assumed to be a copy. There are currently three different "patterns" of zeros written with these E7 bytes, and the pattern is specified by parameter A3 (LCKPAT).

It is difficult to determine the exact pattern used. However, it is easy to spot this protection scheme with the nibble editor. Look for a sector with some highlighted E7's in it. Then, use the 4X4 encoding chart at the end of this manual to determine the number of this sector. Put the track number and sector number in parameters A1 and A2. Most disks use pattern number 1 (parameter A3), so try this first. If the copy still won't boot, try a different pattern number in parm A3. Note that some disks (such as Crossword Magic 4.0) use multiple sectors protected in this manner with different patterns for each sector.

If you are using parm 78 to duplicate Prolok disks, just set this parm to 1. If any tracks contain a single bad sector, this sector will overlay the sector 180 degrees around the disk. This fools the Prolok protection scheme into thinking it has found a real laser-burned diskette as it reads the first bad sector, writes to the second copy (to see if the error is a bad spot in the media itself), then reads the first copy of the sector again. Since it thinks it has written this sector (which would clear up an electronic error), it believes the backup disk is real as it doesn't know it has been fooled by a duplicate copy of this sector.

PARAMETER 79 (KARATE) - BITCOPY ONLY

This parameter was developed for the programs "Karateka" and "Bank Street Speller" by Broderbund, and is one of the most complicated parameters in Copy II Plus. The primary difficulty in making a functional backup of these disks is that there are no obvious sync bytes on most tracks and the track prolog bytes change from track to track. This prevents the standard "Header" and "Sync" track-start finding routines from working.

The way these disks boot is to read a table of headers out of a certain sector which is written in 4X4 encoded format. The placement of this sector can be on any track (including half tracks). Further, the placement within this sector of the table of header bytes (track start bytes) is in different places on different disks. If this weren't enough, the header table is indexed in a rather odd way by the track number being read and the indexing method is different between the two disks!

To complicate matters even more, every 256th byte of the data on these tracks must be written as a 36 microsecond byte (9 bits - one extra zero) in order for the data to be read properly as the data read routines on the original disk are slow when crossing a page boundary.

To copy this disk, the parm "Karate" is set to a non-zero value. If it is a "1", then a Karateka disk is assumed. If it is a "4", then a Bank Street Speller disk is assumed. It needs to know this to use the right indexing method into the track header table.

The next parm, number 7A (KTRACK) is the track number (in quarter tracks) where the sector containing the header table can be found. The parm numbers 7B-7D specify a three-byte header to locate the start of this special sector. (This sector has no ID - it begins with the prolog and the bytes immediately following are the 4X4 encoded (FM) data bytes.)

When the 5.25 bit copy program sees a non-zero value in the "KARATE" parm, it goes into a "build" mode whereby the head is stepped to the specified track, and the header (track start) table is read and decoded from the special sector. It then puts the first track's prolog into ADPRO1-ADPRO2. When this is done, it steps back out to the track to be copied and sets an internal flag that says the build mode is complete so that for future tracks, it will just look up the prolog bytes from the table it has just created from the special sector. It can now find the proper start of each track just like the original program can - it looks it up in the table it read from the master diskette. For each track to be copied, the prolog bytes are first looked up from the table and put into ADPRO1-ADPRO2. The track is then read, then written, and it goes on to the next track, getting the prolog bytes from the table.

Between the read and write phase, another routine is called to set every 256th byte in the buffer as sync so that it can be read properly by the original programs slow data read routine.

PARAMETER 7E, (DZLDRAW) - BITCOPY ONLY

This routine is used to write a field of special sector ID's (sector address fields without any accompanying data). Each ID consists of 3 prolog bytes, then a 4X4 encoded byte, then 3 epilog bytes

followed by 2 zero bytes. This protection scheme works by knowing that the zero bytes on the original disk that separate all these ID's will be read unreliably (the drive will add 1 bits, causing the copy to be more reliably read but in the wrong way). Thus, it is not possible to copy this field of ID's with a standard bit-copy program. This routine writes this field on the specified track so that it can be found.

The routine is enabled by putting a 1 in parameter 7E. Specify the prolog bytes in parms 98-9A (RG1-RG3). Specify the epilog bytes in parms 9B-9D (RG4-RG6). (Note: The "RG" name for these parms are the initials of the programmer who developed it, who was so proud he put his name on each protected disk.)

PARAMETER 7F (SPELL) - BITCOPY ONLY

This parameter will cause a blank track of all FF bytes (not sync) to be written. The read data will be ignored - this is a write only parm. Some disks check for a track of all FF's with a set number of non-FF bytes. Since there are no sync bytes on the original track, you can't specify the track start by using a single byte as it can be read as several possible bytes depending upon how out-of-sync the track is read.

To use this routine, set parameter 7F to the number of non-FF bytes you want to appear (including the track splice) on the target disk. It will write this many AA bytes separated by 256 standard FF bytes. This scheme first appeared on Sensible Speller, which is where it got its name.

PARAMETERS 80-97 (OUTSYNC) - SECTOR COPY ONLY

These parameters are used to write a standard data field a specified number of bits out-of-sync. The sector's address field and data prolog bytes are written normally. Then 1 to 4 "extra"

data prolog bytes are written. During one of these, the disk controller's data latch is reset, causing only some of the bits of that extra data prolog byte to be sent to the disk. The original disk, when reading these special sectors will do the same reset of the latch when reading the data, and expect to get a good sector when these bits are discarded.

Here are how each of these parameters work:

OUTSYN (parm #80)– enables this routine and sets the number of microseconds out–of–sync to write the data. The following table describes these values:

FF	don't write or read out of sync (default)
00	write 16 usec out of sync (discard 4 bits)
01	write 20 usec out of sync (discard 5 bits)
02	write 24 usec out of sync (discard 6 bits)
03	write 28 usec out of sync (discard 7 bits)

Note: The low nibble is used to set the number of bits out–of–sync to write. If you need to specify a different number of bits to read out–of–sync (sector copy does a read–after–write verify), then put it in the high nibble.

OSBYTE (parm #81) – specifies which extra data prolog byte to reset the latch when writing.

DTPRO4 (parm #82) – 1st extra (4th total) data prolog byte.
DTPRO5 (parm #83) – 2nd extra (5th total) data prolog byte.
DTPRO6 (parm #84) – 3rd extra (6th total) data prolog byte.
DTPRO7 (parm #85) – 4th extra (7th total) data prolog byte.
DTPRO8 (parm #86) – 5th extra (8th total) data prolog byte.

NUMSEC (parm #87). Some disks that use this scheme will mix standard sectors with out–of–sync sectors on the same track. This parm will tell sector copy how many out–of–sync sectors to write (the standard sectors must be copied on a separate pass for this track).

SECTBLE (parm #88–97). This is the table of sector numbers to write out–of–sync. The number of sectors in this table is specified in parm NUMSEC (see above).

PARAMETER 9E (HDRTRK) – SECTOR COPY ONLY

This parameter, when set to 1, will cause the first and third address prolog bytes (ADPRO1, ADPRO3) to be logically OR'd with the current track number. This allows sector copy to be used to backup Math Blaster (and others) that require one of the address epilog bytes to be sync, but use variable address prologs.

PARAMETER 9F (ULTFLG) – BITCOPY ONLY

This parameter, when set to 1, will set the ninth byte after the address prolog to sync and set the immediately following byte to an invalid byte ($80). It looks for all occurrences of the currently set address prolog bytes (usually D5 AA 96), then looks 9 bytes ahead and checks this byte for the value stored in parm A0 (ULTVAL). It then sets this byte as sync and replaces the following byte with $80. Used for Ultima IV (and others), however, it is more reliable to backup these disks with sector copy, putting appropriate bytes into the sector epilog parms.

PARAMETER A4 (CHGBYTE) – BITCOPY ONLY

These parameters are used to replace bytes in the bitcopy buffer with other bytes. It can be used to overlay up to 3 bytes into the buffer at a specific offset – usually to put in an invalid byte, but it can also put checked–for bytes after the track has been desynchronized by invalid bytes. It is used by King's Quest, Master Type, and several others.

This routine looks for the pattern specified in CHGPAT, of length CHGLEN, then adds an offset stored in CHGLOW and CHGHI. It then replaces existing bytes with the bytes in CHGBYT1–3. If any of these bytes are zero, it does not replace the existing byte.

CHGBYT1 (parm # A4)	first byte to overlay
CHGBYT2 (parm # A5)	second byte to overlay
CHGBYT3 (parm # A6)	third byte to overlay
CHGLOW (parm #A7)	low offset address
CHGHI (parm #A8)	hi offset address
CHGPAT (parm #a9)	pattern to look for (9 bytes)
CHGLEN (parm #b2)	length of above pattern to check

PARAMETER BE (LONGD5) – SECTOR COPY ONLY

This parameter, when set to 1, will cause the first address prolog byte (ADPRO1) to be written as a 36 usec (9–bit) byte on all even numbers sectors, and as a 40 usec (10–bit) byte on all odd numbers sectors. The default is 0, so normally, all ADPRO1 bytes will be written as 32 usec (8–bit) bytes.

Appendix D – Summary of Parameters

Here is a summary of all the Bit Copy parameters. The parameter number is listed first, followed by the original (or "default") 5.25" value for the parameter (a few of the 3.5" values are different), the parameter name we've given, and a brief description of what the parameter is for. A few parameter numbers are blank. These are parameters that were used in earlier versions of Copy II Plus, but are no longer needed.

Parm Num.	Orig. Value	Parm Name	Description
00	01	EREAD	Number of read retries if track can't be analyzed.
01	01	EWRITE	Number of write retries if write verify fails.
02	1A	TRKMAX	Maximum track length in pages (for error checking).
03	10	TRKMIN	Minimum track length in pages.
04	--		
05	--		
06	01	SYNC.#	Minimum number of sync to constitute a valid sync field for Standardize Sync routine.
07	7F	STAND	Standardized sync value to replace with, for Fix Invalid Nibbles, Clean Sync Fields, and Standardize Sync.
08	00	STANDF	Use Standardize Sync routine? 1=yes, 0=no.
09	00	CLEAN?	Use Clean Sync Fields routine? 1=yes, 0=no.

Parm Num.	Orig. Value	Parm Name	Description
0A	0B	MATCH	Number of bytes to match with ADDRESS.START table when finding track start by header.
0B	01	DISPLAY	01=see track display when copying, 02=enter nibble editor each track, 00=no display.
0C	00	SYNC.TRACK	Track to synchronize to with Synchronize Tracks routine.
0D	00	DOSYNC	Synchronize tracks? 1=yes, 0=no. This is also set by SYNCHRONIZE TRACKS question.
0E	D5	ADDRESS.START	Table of bytes to match
OF	AA		with when finding track
10	96		start by header. Zero
11	00		bytes match anything.
12	00		
13	00		
14	00		
15	AA		
16	AA		
17	00		
18	00		
19	DE	ADDRESS.END	Bytes to match in Clean
1A	AA		Sync Fields.
1B	00		
1C	D5	DATA.START	Bytes to match in Clean
1D	AA		Sync Fields.
1E	AD		
1F	DE	DATA.END	Bytes to match in Clean
20	AA		Sync Fields.
21	00		

Parm Num.	Orig. Value	Parm Name	Description
22	D5	SYNC.START	Bytes to match on reference track in Synchronize Tracks. 3.5" Bit Copy: all bytes must be filled in.
23	AA		
24	96		
25	00		
26	00		
27	00		
28	00		
29	AA		
2A	AA		
2B	00		
2C	00		
2D	DE		
2E	AA		
2F	00		
30	0B	SYNC.MATCH	Number of bytes on reference track to match with SYNC.START table in Synchronize Tracks routine.
31	01	FIX.INVALID?	Use Fix Invalid Nibbles routine? 1=yes, 0=no.
32	02	GLITCH.SIZ	Number of consecutive non-sync bytes that are allowed in a sync field, for Standardize Sync routine.
33	01	CHANGE	In Standardize Sync routine, convert non-sync bytes to sync also? 1=yes, 0=no.
34	00	BIT.FLAG	Use Bit Insert routine? 1=yes, 0=no.
35	DE	BIT.TABLE	Table of bytes to match with for Bit Insert routine.
36	AA		
37	6B		
38	00		
39	00		

Parm Num.	Orig. Value	Parm Name	Description
3A	04	END.GLITCH	Maximum number of consecutive non-sync bytes that are allowed in the last sync field before track start.
3B	--		
3C	--		
3D	0C	KEEP	Number of bytes to shorten all sync fields to, in Track Chop routine.
3E	01	BITS	Number of zero bits to add to all sync bytes when writing.
3F	--		
40	--		
41	--		
42	--		
43	00	PAGE.OVF	Ignore sync fields longer than 256 bytes when looking for track start? 1=yes, 0=no.
44	00	CUT?	Cut track end off a fixed number of bytes from track start? 1=yes, 0=no.
45	08	CUT.HIGH	High byte: Number of bytes to cut from track start.
46	1F	CUT.LOW	Low byte: Number of bytes to cut from track start.
47	01	VERIFY?	Verify track after writing? 1=yes, 0=no.
48	01	PRSLOT	Printer slot number, for printing track buffer or parameter entry.

Parm Num.	Orig. Value	Parm Name	Description
49	--		
4A	3B	PLINE	Number of lines per page to print when printing track buffer.
4B	00	DONIB?	Do nibble counting? 1=yes, 0=no. This is also set by KEEP TRACK LENGTH question.
4C	01	TOLERANCE	How closely (number of bytes) nibble count must match.
4D	01	ERASE	Erase entire track to 32 usec $FF's before writing track data? 1=yes, 0=no.
4E	--		
4F	00	SDFLTR	Don't allow track data to contain bytes with more than 1 consecutive zero? (Continue analyzing?) 1=yes, 0=no.
50	0B	EMATCH	Number of bytes to match to find repeat of track start.
51	0	DYNAM	Do Dynamic Header Change? 1=yes, 0=no.
52	07	DYNAM.LOW	Low byte: Number of bytes from track start to find new header.
53	08	DYNAM.HIGH	High byte: Number of bytes from track start to find new header.
54	01	FILL.ORDER	Fill in header backwards (1) or forwards (0).

Parm Num.	Orig. Value	Parm Name	Description
55	01	FIND.START	Find track start by (2) sync then header, or (3) header then sync.
56	00	OLD.READ	Use old Read Track routine that does not detect sync? 1=yes, 0=no.
Parameters $57 through $77 are used when sector copying a disk.			
57 58 59	D5 AA 96	APRO	Address prolog bytes to match.
5A	00	ASEED	Checksum seed for address field.
5B	FF	ACHKF	Check for address field checksum error? FF=yes, 00=no.
5C 5D 5E 5F	DE AA EB FF	AEPI	Wanted address epilog bytes. Match epilog read against first two of these.
60	FF	AEPIF	Address epilog flag: Check epilogs when reading? Use wanted epilog bytes rather than read epilog bytes when writing? FF=yes, 00=no.
61 62 63	D5 AA AD	DPRO	Data prolog bytes to match.
64	00	DSEED	5.25" Bit Copy: Checksum seed for data field.
64	00	TRKFLG	3.5" Bit Copy: Require correct track and side values in address field? FF=yes, 0=no.

Parm Num.	Orig. Value	Parm Name	Description
65	FF	DCHKF	Check for data field checksum error? FF=yes, 00=no.
66	DE	DEPI	Data epilog bytes to match.
67	AA		
68	EB		
69	FF		
6A	FF		
6B	FF	DEPIF	Data epilog flag: Check epilogs when reading? Use wanted epilog bytes rather than read epilog bytes when writing? FF=yes, 00=no.
6C	93	FUNNY	5.25" Bit Copy: Funny sync bytes to write before address field.
6D	F3		
6E	FC		
6F	FF		
70	FF		
71	02	TIME	5.25" Bit Copy: Number of zero bits to add to each FUNNY byte when writing.
72	02		
73	01		
74	02		
75	02		
76	00	FNYFLG	5.25" Bit Copy: Write FUNNY bytes rather than the last 5 sync $FF's before each address field? FF=yes, 00=no
77	00	DOSFLG	5.25" Bit Copy: Try copying DOS 3.3 only, rather than trying DOS 3.2 first? FF=yes, 00=no.
6C	00	DTSEED1	3.5" Bit Copy: Checksum seeds for data field.
6D	00	DTSEED2	
6E	00	DTSEED3	

Parm Num.	Orig. Value	Parm Name	Description
78-BE	--		Special, see "Special Parms" in Appendix C.
FF	--	RESTORE	If you access this special parameter manually, it restores all parameters back to their original values.

Appendix E – Number Conversion Tables

The table below lets you convert between decimal, hexadecimal, and binary numbers. A thorough tutorial on number systems is beyond the scope of this manual. Suffice it to say that decimal (base 10), hexadecimal (base 16), and binary (base 2) simply provide different ways of expressing any number. Decimal 11 is exactly the same as hex $0B and binary 00001011. A single hex digit is called a "nibble" or "nybble"; a binary digit is a "bit". Many computer concepts and disk values can be expressed more readily using hex or binary than with decimal. That's why Copy II Plus uses hexadecimal numbers for some values.

The fourth column includes the corresponding Apple 5.25" disk 4-and-4 encoded values (in hexadecimal) for each number. The last column includes the corresponding 6-and-2 translated valid disk values (in hexadecimal). There are only 64 translated values possible, so this column does not continue for the entire table. (Since not all of the 256 possible byte values can be stored reliably on a disk, some form of encoding or translation to valid disk bytes is always necessary before the information is written to disk. See Appendix A for how 4-and-4 encoded numbers and 6-and-2 translated numbers are used on a disk.)

Dec	Hex	Binary	4-and-4	Translated
0	$00	00000000	AA AA	96
1	$01	00000001	AA AB	97
2	$02	00000010	AB AA	9A
3	$03	00000011	AB AB	9B
4	$04	00000100	AA AE	9D
5	$05	00000101	AA AF	9E
6	$06	00000110	AB AE	9F
7	$07	00000111	AB AF	A6
8	$08	00001000	AE AA	A7
9	$09	00001001	AE AB	AB
10	$0A	00001010	AF AA	AC
11	$0B	00001011	AF AB	AD
12	$0C	00001100	AE AE	AE
13	$0D	00001101	AE AF	AF

Dec	Hex	Binary	4-and-4	Translated
14	$0E	00001110	AF AE	B2
15	$0F	00001111	AF AF	B3
16	$10	00010000	AA BA	B4
17	$11	00010001	AA BB	B5
18	$12	00010010	AB BA	B6
19	$13	00010011	AB BB	B7
20	$14	00010100	AA BE	B9
21	$15	00010101	AA BF	BA
22	$16	00010110	AB BE	BB
23	$17	00010111	AB BF	BC
24	$18	00011000	AE BA	BD
25	$19	00011001	AE BB	BE
26	$1A	00011010	AF BA	BF
27	$1B	00011011	AF BB	CB
28	$1C	00011100	AE BE	CD
29	$1D	00011101	AE BF	CE
30	$1E	00011110	AF BE	CF
31	$1F	00011111	AF BF	D3
32	$20	00100000	BA AA	D6
33	$21	00100001	BA AB	D7
34	$22	00100010	BB AA	D9
35	$23	00100011	BB AB	DA
36	$24	00100100	BA AE	DB
37	$25	00100101	BA AF	DC
38	$26	00100110	BB AE	DD
39	$27	00100111	BB AF	DE
40	$28	00101000	BE AA	DF
41	$29	00101001	BE AB	E5
42	$2A	00101010	BF AA	E6
43	$2B	00101011	BF AB	E7
44	$2C	00101100	BE AE	E9
45	$2D	00101101	BE AF	EA
46	$2E	00101110	BF AE	EB
47	$2F	00101111	BF AF	EC
48	$30	00110000	BA BA	ED
49	$31	00110001	BA BB	EE
50	$32	00110010	BB BA	EF
51	$33	00110011	BB BB	F2
52	$34	00110100	BA BE	F3
53	$35	00110101	BA BF	F4
54	$36	00110110	BB BE	F5
55	$37	00110111	BB BF	F6
56	$38	00111000	BE BA	F7

Dec	Hex	Binary	4-and-4	Translated
57	$39	00111001	BE BB	F9
58	$3A	00111010	BF BA	FA
59	$3B	00111011	BF BB	FB
60	$3C	00111100	BE BE	FC
61	$3D	00111101	BE BF	FD
62	$3E	00111110	BF BE	FE
63	$3F	00111111	BF BF	FF
64	$40	01000000	AA EA	
65	$41	01000001	AA EB	
66	$42	01000010	AB EA	
67	$43	01000011	AB EB	
68	$44	01000100	AA EE	
69	$45	01000101	AA EF	
70	$46	01000110	AB EE	
71	$47	01000111	AB EF	
72	$48	01001000	AE EA	
73	$49	01001001	AE EB	
74	$4A	01001010	AF EA	
75	$4B	01001011	AF EB	
76	$4C	01001100	AE EE	
77	$4D	01001101	AE EF	
78	$4E	01001110	AF EE	
79	$4F	01001111	AF EF	
80	$50	01010000	AA FA	
81	$51	01010001	AA FB	
82	$52	01010010	AB FA	
83	$53	01010011	AB FB	
84	$54	01010100	AA FE	
85	$55	01010101	AA FF	
86	$56	01010110	AB FE	
87	$57	01010111	AB FF	
88	$58	01011000	AE FA	
89	$59	01011001	AE FB	
90	$5A	01011010	AF FA	
91	$5B	01011011	AF FB	
92	$5C	01011100	AE FE	
93	$5D	01011101	AE FF	
94	$5E	01011110	AF FE	
95	$5F	01011111	AF FF	
96	$60	01100000	BA EA	
97	$61	01100001	BA EB	
98	$62	01100010	BB EA	
99	$63	01100011	BB EB	

Dec	Hex	Binary	4-and-4
100	$64	01100100	BA EE
101	$65	01100101	BA EF
102	$66	01100110	BB EE
103	$67	01100111	BB EF
104	$68	01101000	BE EA
105	$69	01101001	BE EB
106	$6A	01101010	BF EA
107	$6B	01101011	BF EB
108	$6C	01101100	BE EE
109	$6D	01101101	BE EF
110	$6E	01101110	BF EE
111	$6F	01101111	BF EF
112	$70	01110000	BA FA
113	$71	01110001	BA FB
114	$72	01110010	BB FA
115	$73	01110011	BB FB
116	$74	01110100	BA FE
117	$75	01110101	BA FF
118	$76	01110110	BB FE
119	$77	01110111	BB FF
120	$78	01111000	BE FA
121	$79	01111001	BE FB
122	$7A	01111010	BF FA
123	$7B	01111011	BF FB
124	$7C	01111100	BE FE
125	$7D	01111101	BE FF
126	$7E	01111110	BF FE
127	$7F	01111111	BF FF
128	$80	10000000	EA AA
129	$81	10000001	EA AB
130	$82	10000010	EB AA
131	$83	10000011	EB AB
132	$84	10000100	EA AE
133	$85	10000101	EA AF
134	$86	10000110	EB AE
135	$87	10000111	EB AF
136	$88	10001000	EE AA
137	$89	10001001	EE AB
138	$8A	10001010	EF AA
139	$8B	10001011	EF AB
140	$8C	10001100	EE AE
141	$8D	10001101	EE AF
142	$8E	10001110	EF AE

Dec	Hex	Binary	4-and-4
143	$8F	10001111	EF AF
144	$90	10010000	EA BA
145	$91	10010001	EA BB
146	$92	10010010	EB BA
147	$93	10010011	EB BB
148	$94	10010100	EA BE
149	$95	10010101	EA BF
150	$96	10010110	EB BE
151	$97	10010111	EB BF
152	$98	10011000	EE BA
153	$99	10011001	EE BB
154	$9A	10011010	EF BA
155	$9B	10011011	EF BB
156	$9C	10011100	EE BE
157	$9D	10011101	EE BF
158	$9E	10011110	EF BE
159	$9F	10011111	EF BF
160	$A0	10100000	FA AA
161	$A1	10100001	FA AB
162	$A2	10100010	FB AA
163	$A3	10100011	FB AB
164	$A4	10100100	FA AE
165	$A5	10100101	FA AF
166	$A6	10100110	FB AE
167	$A7	10100111	FB AF
168	$A8	10101000	FE AA
169	$A9	10101001	FE AB
170	$AA	10101010	FF AA
171	$AB	10101011	FF AB
172	$AC	10101100	FE AE
173	$AD	10101101	FE AF
174	$AE	10101110	FF AE
175	$AF	10101111	FF AF
176	$B0	10110000	FA BA
177	$B1	10110001	FA BB
178	$B2	10110010	FB BA
179	$B3	10110011	FB BB
180	$B4	10110100	FA BE
181	$B5	10110101	FA BF
182	$B6	10110110	FB BE
183	$B7	10110111	FB BF
184	$B8	10111000	FE BA
185	$B9	10111001	FE BB

Dec	Hex	Binary	4-and-4
186	$BA	10111010	FF BA
187	$BB	10111011	FF BB
188	$BC	10111100	FE BE
189	$BD	10111101	FE BF
190	$BE	10111110	FF BE
191	$BF	10111111	FF BF
192	$C0	11000000	EA EA
193	$C1	11000001	EA EB
194	$C2	11000010	EB EA
195	$C3	11000011	EB EB
196	$C4	11000100	EA EE
197	$C5	11000101	EA EF
198	$C6	11000110	EB EE
199	$C7	11000111	EB EF
200	$C8	11001000	EE EA
201	$C9	11001001	EE EB
202	$CA	11001010	EF EA
203	$CB	11001011	EF EB
204	$CC	11001100	EE EE
205	$CD	11001101	EE EF
206	$CE	11001110	EF EE
207	$CF	11001111	EF EF
208	$D0	11010000	EA FA
209	$D1	11010001	EA FB
210	$D2	11010010	EB FA
211	$D3	11010011	EB FB
212	$D4	11010100	EA FE
213	$D5	11010101	EA FF
214	$D6	11010110	EB FE
215	$D7	11010111	EB FF
216	$D8	11011000	EE FA
217	$D9	11011001	EE FB
218	$DA	11011010	EF FA
219	$DB	11011011	EF FB
220	$DC	11011100	EE FE
221	$DD	11011101	EE FF
222	$DE	11011110	EF FE
223	$DF	11011111	EF FF
224	$E0	11100000	FA EA
225	$E1	11100001	FA EB
226	$E2	11100010	FB EA
227	$E3	11100011	FB EB
228	$E4	11100100	FA EE

Dec	Hex	Binary	4-and-4
229	$E5	11100101	FA EF
230	$E6	11100110	FB EE
231	$E7	11100111	FB EF
232	$E8	11101000	FE EA
233	$E9	11101001	FE EB
234	$EA	11101010	FF EA
235	$EB	11101011	FF EB
236	$EC	11101100	FE EE
237	$ED	11101101	FE EF
238	$EE	11101110	FF EE
239	$EF	11101111	FF EF
240	$F0	11110000	FA FA
241	$F1	11110001	FA FB
242	$F2	11110010	FB FA
243	$F3	11110011	FB FB
244	$F4	11110100	FA FE
245	$F5	11110101	FA FF
246	$F6	11110110	FB FE
247	$F7	11110111	FB FF
248	$F8	11111000	FE FA
249	$F9	11111001	FE FB
250	$FA	11111010	FF FA
251	$FB	11111011	FF FB
252	$FC	11111100	FE FE
253	$FD	11111101	FE FF
254	$FE	11111110	FF FE
255	$FF	11111111	FF FF

Appendix F – Extra Memory

Both the Utilities COPY DISK option and the 3.5" Bit Copy program can take advantage of extra memory in your computer so that fewer disk swaps are needed for the copy. An internal Copy II Plus "Memory Manager" section within both programs invisibly handles this extra memory.

Below is a brief discussion of the Copy II Plus Memory Manager. You don't need to know this information to use Copy II Plus. It's provided here for your reference only.

There are several different kinds of memory that can be added to an Apple or Apple-compatible computer: expansion RAM in the Apple IIGS, numbered-slot Apple style (or Laser 128EX) Memory Expansion RAM, and auxiliary-slot RAM boards for the Apple IIe or IIc. Unfortunately, all memory is not created equal! The extra memory does not appear to the computer as just "more main memory". Each kind of expansion memory appears in a different way and requires a different method to access. This means that for an application program to use the various kinds of memory, *additional programming must be written to handle each*. The Copy II Plus Memory Manager does this handling.

The Memory Manager can access extra memory in two ways:
1. It can make use of available expansion memory in an Apple IIGS.
2. If there are one or more ProDOS-compatible RAMdisks available, it will find the one RAMdisk that has the most free space, and use that remaining space without disturbing other files on the RAMdisk.

With those two methods, it can get memory from:

- Apple IIGS expansion memory, whether or not that memory is set up (from the Control Panel) as a RAMdisk.

- a numbered-slot memory board, such as the Apple 1 Meg Memory Expansion Board, the Applied Engineering RamFactor board, or the AST SprintDisk.
- the slot 5 memory expansion in a Laser 128 EX.
- the slot 4 memory expansion in a newer Apple IIc if equipped with IIc Memory Expansion Card.
- an Apple IIe (or IIc) auxiliary-memory board if it has already been set up as a RAMdisk, such as the Applied Engineering RamWorks boards set up with their "ProDrive" RAMdisk software, or the CheckMate Technology MultiMate board set up with their RAMdisk software. (See the section on Large Memory Boards at the end of Chapter Two.)

The Memory Manager section of course takes up memory itself! It must reside in the main memory of the computer in order to work with the other kinds of memory. This leaves less room in main memory for the rest of the program and for data areas. The saying "You need money to make money" could be adapted as "You need memory to get memory". For this reason:

- In the Utilities COPY DISK option, the Memory Manager requires at least 128K of memory built into the computer in order to access additional memory. Therefore, it cannot find additional memory in an Apple II or Apple II Plus (or II Plus compatible) computer. Also, if ProDOS's small slot 3 drive 2 RAMdisk is used or formatted, this ties up the additional memory, so the Memory Manager is no longer available until you restart Copy II Plus.

- The 3.5" Bit Copy program requires 128K of memory to work at all. With this space, the Memory Manager is always available to access additional memory.

- The 5.25" Bit Copy program is designed to work entirely in 64K RAM, so the Memory Manager is not included.

In either COPY DISK or in the 3.5" Bit Copy program, if the Memory Manager has trouble using the memory (for example if a

RAMdisk is not working correctly), it will display a message letting you know of the problem. If you copy again, the Memory Manager won't be used this time, so that you can continue with the copy.

Appendix G – Other Products

Central Point Software also sells these other disk utility and software backup products:

COPY II PC is the most complete copy program available; it's clever enough to outwit most of the newest and most advanced protection schemes available! It can also run popular software such as Lotus 1-2-3 from the hard disk without inserting the original , and eliminates the need to uninstall before doing a hard disk backup and restore. **Copy II PC** includes a disk drive speed test to help keep your drives in top shape. For IBM PC/XT/AT; PS/2 Models 30, 50, 60 and 80; some compatibles. $39.95 plus $3 shipping, $8 outside North America.

COPY II PC DELUXE OPTION BOARD
Copy II PC can back up most protected software for the IBM. (It can even run quite a few of them from your hard disk without a floppy in drive A.) But it can't back up everything. So we created the **Deluxe Option Board, hardware that lets you make floppy (only) backups of almost every program available for the IBM, as well as transfer IBM/Mac data files!**

The Option Board uses the same disk duplication technology used by software duplication firms who put the protection on in the first place. There us virtually no protection scheme the Option Board cannot handle, (except those "protected" by physically altering the disk).

The Deluxe Option Board also makes it easy for **PC's to trade data files with Macintoshes!** With the Copy II PC Deluxe Option Board, your computer's internal 3.5" drive is transformed into a dual purpose Mac/IBM compatible drive for hundreds less!

The Deluxe Option Board will not interfere with any other hardware or software and is not needed to run the backups, only to create them.

Copy II Plus Disk Utilities

Now supports 1.2 and 1.44 Meg drives! The Deluxe Option Board works on IBM PC/XT/AT, PS/2 Models 25 ad 30**; Zenith 150, 151, 158; Compaq Deskpro, 286 Plus*, Portable**; 256K Tandy 1000*, 1000SX, 1000TX* **. Requires one slot.
* These computers require an extra $15 cable
** These computers require specific installation procedures.
Please specify when ordering. Just $159 plus $5 shipping, $15 outside U.S.

PC TOOLS Deluxe is our complete DOS utility package for the IBM PC and nearly all compatible computers. PC TOOLS Deluxe combines all the popular features of the Norton Utilities, Fastback, XTREE, Mace, Sidekick and Disk Optimizer. When it comes to managing and protecting your data, nothing does it better than PC Tools Deluxe. It includes:

- The best **UNDELETE** available–recovering all data on even fragmented files

- Hard disk backup faster and every bit as reliable as Fastback

- The leading **UNFORMAT** for hard disks and floppy disks, enabling you to recover from almost any disk disaster

- **PCFORMAT** for 100% safe formatting of floppy and hard disks

- Fast, reliable **DISK CACHING** for speeding up disk access

- A speedy **COMPRESS** feature that like Disk Optimizer dramatically improves hard disk performance

- A better resident mini **WORD PROCESSOR** than Sidekick, including word wrap, search/replace, formatted page printing and more

- A complete **DOS SHELL** that lets you access DOS even inside other programs and includes a graphic directory display

PC TOOLS Deluxe works with virtually all IBM compatibles with at least 256K of memory. ($79.00)

COPY II for the Apple Macintosh is a disk backup and utility package for Macintosh computers. Combining all the most needed functions, Copy II will let you make archival backups of your protected software easily. No parameters are needed. The new Track-Level Editor allows you to spot copy protections, repair damaged blocks, etc. A smart Locate feature finds files anywhere on your drives. Copy II also includes our powerful utility program "MacTools". MacTools can recover some damaged disks and undelete files. It will also display all files on a disk, including invisible invisible files, and will let you make them visible again. You can mark any file as protected, unprotected, locked or unlocked. Its block editor shows you what is inside any Macintosh file, and allows you to change it. Our "COPY II HARD DISK" application, also included on the disk, can copy many popular protected programs onto a hard disk. ($39.95)

COPY II 64/128 is our disk copy program for the Commodore 64 and 128 computers. Copy II 64/128 makes reliable backups of nearly all copy-protected Commodore software. It's fast – it can backup disks in 2 minutes (single drive) or 1 minute (dual drives) – and it can handle numerous protection schemes automatically. Copy II 64 fully supports the Commodore 128 and 1571 drives, and uses all the memory in the Commodore 128 computer to minimize disk swapping if only one disk drive is used. Other features of Copy II 64/128 include the ability to delete files, format disks (single and double sided in as little as 15 seconds) and a quick loader to make loading other programs as fast as possible.

We update our software products regularly to handle new protections; updates are always available to you as a registered owner at a reduced price. Protection schemes do change frequently so it's a good idea to double check with us if you

to back up a brand new release of one particular program. For more information on our products, call or write:

Central Point Software Inc.
15220 NW Greenbrier Parkway, Suite 200
Beaverton, OR 97006
(503)690–8090

Index

"Copy II Plus wiped, reformatted, and deleted its competition."
—inCider

■ **Award-winning utilities.** Over 20 built-in utilities, including the ability to delete and undelete files, map disk usage, copy disks and files, view and print files, format and verify disks, and alphabetize the catalog. Also contains drive diagnostics for checking and adjusting drive speed. Plus, much more.

■ **Versatile performer–handles both DOS formats.** Copy II Plus enables you to convert DOS 3.3 files to ProDOS files and vice versa–automatically.

■ **Supports IIGS and Apple 1-megabyte RAM boards.** Copy II Plus uses available memory to minimize disk swaps during disk copying in utilities and bit copy programs.

■ **If you're really into what's on your disk.** The Copy II Plus Sector Editor allows you to directly view and change the data in any sector or block. Poke around in HEX or ASCII format, disassemble and list code, or use the Nibble Editor to see actual bytes stored on any track of the disk.

■ **Protect your software investment.** Make backup copies of protected and unprotected programs. Central Point's state-of-the-art technology enables you to duplicate many programs–even ones with the most sophisticated protection schemes. Use Copy II and you'll always have a backup handy should anything happen to your original disk.

■ **Easy to use.** Backup parameters for most programs are already on the disk. Simply select which program you want to backup, insert that program disk and you're off. Data and unprotected disks are copied in less than a minute and require only two passes on an Apple //c or 128K //e.

■ **Everything you always wanted to know, but were(n't) afraid to ask.** Whether you're a novice or an experienced computer buff, you'll find the Copy II Plus manual easy to read, easy to follow, and, most of all, easy to understand.

Hardware requirements:
Apple II or Laser computer with 128K memory and one disk drive.

3.5-inch bit copy requires Laser 128EX or 128EX/II with a Laser 3.5-inch drive, or Apple //c Plus or IIGS with the Apple IIGS or Laser 3.5-inch drive, or Apple II with Central Point Universal Disk Controller and Laser 3.5-inch drive. Apple **Unidisk** 3.5-inch drives are not supported.

Central Point Software
INCORPORATED
15220 N.W. Greenbrier Parkway, #200
Beaverton, OR 97006
(503) 690-8090

15M/WCP/0189