

Hardcore

# COMPUTIST

Issue No. 16

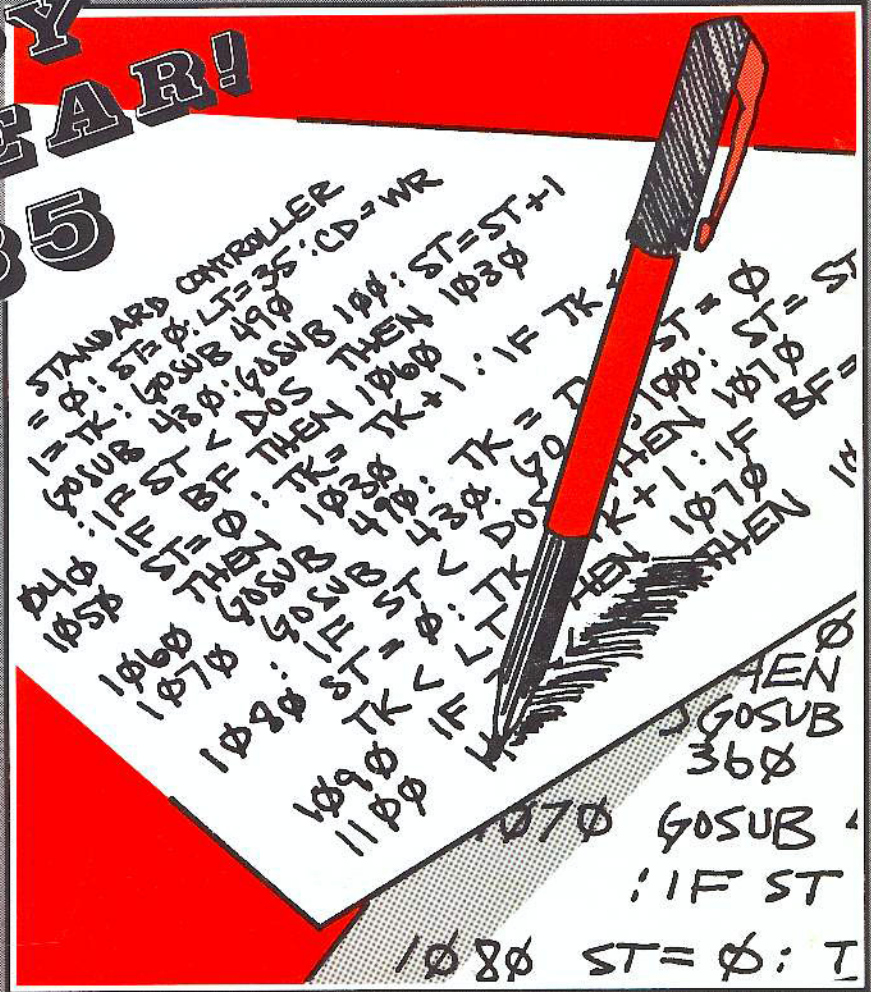
\$2.50

**HAPPY  
NEW YEAR!  
1985**

**Secret Weapon:  
RAMcard**  
Pg. 9

**Deprotecting  
Sensible Speller  
for ProDOS**  
Pg. 12

**Softkey For  
Sideways**  
Pg. 25



**The Controller Writer**  
Pg. 15

Hardcore COMPUTIST  
PO Box 110846-T  
Tacoma, WA 98411



BULK RATE  
 U.S. Postage  
**PAID**  
 Tacoma, WA  
 Permit No. 269



# TURBO CHARGE YOUR IIe AND EXPAND APPLE WORKS TO 101K

## MEMORYMASTER IIe™ 128K RAM CARD

As an Apple user, you already know all the things your IIe can do. Now Applied Engineering expands that list with MemoryMaster IIe™.

With the MemoryMaster IIe, you'll have up to 192K RAM at your command. Designed expressly for the auxiliary slot in the IIe, the MemoryMaster IIe provides an 80 column video display and up to 128K of memory using just one slot. But the MemoryMaster IIe differs from other large memory cards in one important way, both the 64K and 128K configurations are **TOTALLY** compatible with software written for the Apple IIe 80 column and extended 80 column cards. In fact, it's not possible to write a program that uses the smaller cards from Apple that won't work with the MemoryMaster IIe.

But the MemoryMaster IIe is not just another piece of unsupported

hardware. Each MemoryMaster IIe includes a multi-programming environment program which will enable you to have three different programs ready to run at any moment.

Many of today's software packages for data-base management, word processing, business applications and spread sheets either require or are enhanced by the MemoryMaster IIe. And more and more programs are using double high resolution graphics for which the MemoryMaster IIe is required.

If you already have Apple's 64K card, just order the MemoryMaster IIe with 64K and use the 64K from your old board to give you a full 128K. The board is fully socketed so you simply plug in your chips.

- Expands your Apple IIe to 192K memory.
- Provides an 80 column text display.
- **TOTALLY** compatible with all Apple IIe 80 column and extended 80 column card software—there are **NO** exceptions.
- Available in 64K and 128K configurations.
- The 64K configuration is **USER** upgradeable to 128K.
- Can be used as a solid state disk drive to make your programs run over 20 times faster (the 64K configuration will act as half a drive).
- Bank select LED's for each 64K bank.
- Permits your IIe to use double high resolution graphics.
- Uses the same commands as the Apple 80 column card.
- Plugs into the auxiliary slot in the Apple IIe.
- The 64K MemoryMaster IIe will automatically expand VisiCalc to 95K storage, in 80 columns! The 128K MemoryMaster IIe will expand VisiCalc to 141K storage with optional pre-boot disk.
- The 64K MemoryMaster IIe will automatically expand Apple Works to 55K storage, in 80 columns! The 128K MemoryMaster IIe will expand Apple Works to 101K storage with optional expand disk.
- Fully PASCAL and CP/M compatible.
- Lowest power consuming 128K card available.
- Complete documentation included.
- PRO-DOS will automatically use the MemoryMaster IIe as a high speed disk drive.
- Three year warranty.

### Low Cost Options.

#### Apple Works Expand™

Although Apple Works is compatible with both a 64K and 128K MemoryMaster IIe, Apple Works only "sees" it's first 64K bank. Our Apple Works expand program will make a modification to Apple Works that simply lets it know you've got more memory, giving you 101K work space.

**\$29**

#### Ram Drive IIe™

Sure-fire wait-reduction! Time was, you had to wait for your disk drives. But Ram Drive IIe has changed all that, giving you a large, extremely fast, solid-state disk drive, much faster than floppies or hard disks.

Ram Drive IIe works with either the 64K or 128K MemoryMaster IIe to give you a high speed solid-state disk drive. The Ram Drive IIe software features audio-visual access indicators, easy setup for turnkey operation, and easy menu driven documentation. The program can be modified and is copyable. If you have a 64K MemoryMaster, Ram Drive IIe will act as half a disk drive. If you have a 128K MemoryMaster, Ram Drive IIe will act as a full disk drive. Ram Drive IIe is compatible with APPLESOFT, DOS3.3, PRO-DOS, and PASCAL. Disk also includes a high speed RAM disk copying program. Ram Drive is another disk drive only 20 times faster.

**\$29**

#### CP/M Ram Drive IIe

CP/M Ram Drive IIe is just like the Ram Drive IIe above, only for CP/M. CP/M Ram Drive IIe runs on any Z-80 card that runs standard CP/M i.e. Applied Engineering Z-80 Plus or Microsoft Soft Card. CP/M Ram Drive will dramatically speed up the operation of most CP/M software because CP/M normally goes to disk fairly often. Fast acting software like dBase II and Wordstar become virtually instantaneous when used with CP/M Ram Drive.

**\$29**

#### VC IIe Expander

VC IIe Expander will give owners of VisiCalc IIe and the 128K MemoryMaster a total of 141K work space. This disk will pre-boot VC IIe in 1.5 seconds.

**\$29**

#### Advanced VC IIe Expander

Advanced VC IIe Expander is just like VC IIe Expand only it is designed for Advanced VC IIe. Your work space will be increased to 131K. This disk will pre-boot in 1.5 seconds.

**\$29**

MEMORYMASTER IIe with 128K .....	\$249
Upgradeable MEMORYMASTER IIe with 64K .....	\$169
A. E. Extended 80 Column Card with 64K .....	\$139

Texas Residents Add 5% Sales Tax  
Add \$10.00 If Outside U.S.A.

Send Check or Money Order to:  
APPLIED ENGINEERING  
P.O. Box 798  
Carrollton, TX 75006

Call (214) 492-2027  
8 a.m. to 11 p.m. 7 days a week  
MasterCard, Visa & C.O.D. Welcome  
No extra charge for credit cards.

# DISKBUSTERS ALERT!

Want to show your friends  
just what you think of software COPY-PROTECTION?

Wear the ORIGINAL Hardcore COMPUTIST Diskbusters t-shirt!



## Diskbusters

\*Hardcore COMPUTIST 1984

Only  
**\$9.95**

\* Hanes 50% cotton/ 50% polyester

**Hardcore  
COMPUTIST**

## Join The "OFFICIAL" DISKBUSTERS team!!!

Don't miss this opportunity! Order the ORIGINAL Diskbusters t-shirt TODAY and you'll never again be "just one of the crowd". Let the world know that YOU are a member of the DISKBUSTERS team of Hardcore COMPUTIST, the magazine for the serious user of Apple ][ computers.

Tell your friends! Tell your enemies! (Maybe even tell your folks!) They'll want to know how they, too, can become a member of the team.

**Hardcore COMPUTIST says:  
"We ain't afraid of no disk,"  
and YOU CAN, TOO!**

**ORDER TODAY!**

Use your VISA/MC, personal check or money order

Available in adult sizes only (black & red on gray).

Rush me \_\_\_\_\_ (total) DISKBUSTERS T-shirts in the sizes indicated below. I have enclosed \$9.95 (plus applicable tax & shipping) for each shirt.

ADULT MENS:    \_\_\_ Small    \_\_\_ Med.    \_\_\_ Large    \_\_\_ X-Large

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_

Phone \_\_\_\_\_

VISA/MC    \_\_\_\_\_    \_\_\_\_\_    \_\_\_\_\_    Exp \_\_\_\_\_

Signature \_\_\_\_\_

Send check or money order to: Hardcore T-shirts, PO Box 110816, Tacoma, WA 98411. Washington residents add 7.8% sales tax. Foreign orders add 20% shipping and handling. US funds drawn on US bank.

PO Box 110816  
Tacoma, WA 98411

**M**any of the articles published in Hardcore COMPUTIST detail the removal of copy protection schemes from commercial disks or contain information on copy protection and backup methods in general. We also print bit copy parameters, tips for adventure games, advanced playing techniques (APT's) for arcade game fanatics and any other information which may be of use to the serious Apple user.

Hardcore COMPUTIST also contains a center CORE section which generally focuses on information not directly related to copy-protection. Topics may include, but are not limited to, tutorials, hardware/software product reviews and application and utility programs.

**What Is a Softkey Anyway?** A softkey is a term which we coined to describe a procedure that removes, or at least circumvents, any copy protection that may be present on a disk. Once a softkey procedure has been performed, the disk can usually be duplicated by the use of Apple's COPYA program which is on the DOS 3.3 System Master Disk.

**Following A Softkey Procedure:** The majority of the articles in Hardcore COMPUTIST which contain a softkey will also include a discussion of the type of copy protection present on the disk in question and the technique(s) necessary to remove that protection. Near the end of the article, a step-by-step "cookbook" method of duplicating the disk will appear. Generally, the appropriate actions for the reader to perform will appear in boldface type. Examples are:

1) Boot the disk in slot 6

PR#6

or

2) Enter the monitor

CALL -151

It is assumed that the reader has some familiarity with his or her Apple, i.e. knowing that the RETURN key must be hit following the commands illustrated above.

Hardcore COMPUTIST tries to verify the softkeys which are published, although occasionally this is not possible. Readers should be aware that different, original copies of the same program will not always contain an identical protection method. For this reason, a softkey may not work on the copy of a disk that you own, but it may work on a different copy of the same program. An example of this is Zaxxon, by Datasoft, where there are at least 3 different protection methods used on various releases of the game.

**Requirements:** Most of the programs and softkeys which appear in Hardcore COMPUTIST require an Apple II+ computer (or compatible) with a minimum 48K of RAM and at least one disk drive with DOS 3.3. Occasionally, some programs and procedures have special requirements such as a sector editing program or a "nonautostart" F8 monitor ROM. The prerequisites for deprotection techniques or programs will always be listed at the beginning article under the "Requirements:" heading.

**Software Recommendations:** Although not absolutely necessary, the following categories of utilities are recommended for our readers who wish to obtain the most benefit from our articles:

1) **Applesoft Program Editor** such as Global Program Line Editor (GPLE).

2) **Disk Editor** such as DiskEdit, ZAP from Bag of Tricks or Tricky Dick from The CIA.

3) **Disk Search Utility** such as The Inspector, or The Tracer from The CIA.

4) **Assembler** such as the S-C Macro Assembler or Big Mac.

5) **Bit Copy Program** such as COPY II+, Locksmith or The Essential Data Duplicator.

6) **Text Editor** capable of producing normal sequential text files such as Applewriter II, Magic Window II or Screenwriter II.

Three programs on the DOS 3.3 System Master Disk, COPYA, FID and MUFFIN, also come in very handy from time to time.

**Hardware Recommendations:** Certain softkey procedures require that the computer have some means of entering the Apple's system monitor during the execution of a copy-protected program. For Apple II+ owners there are three basic ways this can be achieved:

1) Place an INTEGER BASIC ROM card in one of the Apple's slots.

2) Install an old monitor or modified F8 ROM on the Apple's motherboard. The installation of a modified F8 ROM is discussed in Ernie Young's, "Modified ROMS", which appeared in Hardcore COMPUTIST No.6.

3) Have available a non-maskable interrupt (NMI) card such as Replay or Wildcard.

Longtime readers of Hardcore COMPUTIST will vouch for the fact that the ability to RESET into the monitor at will, greatly enhances the capacity of the Apple owner to remove copy protection from protected disks.

A 16K or larger RAM card is also recommended for Apple II or II+ owners. A second disk drive is handy, but is not usually required for most programs and softkeys.

**Recommended Literature:** The Apple II and II+ 's come bundled with an Apple Reference Manual, however this book is not included with the purchase of an Apple II/e. This book is necessary reference material for the serious computist. A DOS 3.3 manual is also recommended. Other helpful books include:

**Beneath Apple DOS,** Don Worth and Peter Leichner, Quality Software. \$19.95.

**Assembly Lines: The Book,** Roger Wagner, Softalk Books. \$19.95.

**What's Where In The Apple,** William Lubert, Micro Ink. \$24.95.

**Typing In BASIC Programs:** When typing in basic programs, you will often encounter a delta ("▲") character. These are the spaces you MUST type in if you wish your checksums to match ours. All other spaces are merely printed for easier reading and don't have to be keyed in. Any spaces after the word DATA that aren't delta characters MUST be omitted!

It is a good idea to SAVE your BASIC program to disk frequently while typing it in to minimize the loss of data in the event of a power failure.

**Checksoft:** Checksoft is a Binary program that checks Applesoft programs to ensure that you have keyed them in properly. Every binary program we print has companion checksums which consist of the Applesoft program's line numbers and a hexadecimal (base 16) number for each line. After keying in a BASIC program, BRUN checksoft and compare the checksums for every line that Checksoft generates with those at the end of the program. If you use Checksoft and make a typing error, your checksums will differ from ours beginning at the line where you made the error.

**Typing In Binary Programs:** Binary programs are printed in two different formats, as source code and as object code in a hexadecimal dump. If you want to type in the source code, you will need an assembler. The S-C Macro Assembler is used to generate all the source code which we print. In our source code listings, the memory address of the each instruction is printed at the beginning of every line (instead of the line number).

Binary programs can also be entered directly with the use of the Apple monitor by typing in the bytes listed in the hexdump at the appropriate addresses. Be sure to enter the monitor with a CALL -151 before entering the hexdump. Don't type the checksums printed at the end of each line of the hexdump and don't forget to BSAVE binary programs with the proper address and length parameters listed in the article.

**Checkbin:** Like Checksoft, Checkbin also generates checksums, but was designed to check binary (machine language) programs.

Whenever Hardcore COMPUTIST prints a hexdump to type in, the associated Checkbin generated checksums are printed after every 8 bytes and at the end of every line.

Checksoft and Checkbin were printed in Hardcore COMPUTIST No. 1 and the Best Of Hardcore Computing and are sold on Program Library Disk No. 1 and the Best Of Hardcore Library Disk.

**Let Us Hear Your Likes And Grips:** New and longtime readers of Hardcore COMPUTIST are encouraged to let us know what they like and don't like about our magazine by writing letters to our INPUT column. Our staff will also try to answer questions submitted to the INPUT column, although we cannot guarantee a response due to the small size of our staff. Also, send your votes for the softkeys you would like to see printed to our "Most Wanted List."

---

## How-To's Of Hardcore

Welcome to Hardcore COMPUTIST, a publication devoted to the serious user of Apple II and Apple II compatible computers. We believe our magazine contains information you are not likely to find in any of the other major journals dedicated to the Apple market.

Our editorial policy is that we do NOT condone software piracy, but we do believe that honest users are entitled to back up commercial disks they have purchased. In addition to the security of a backup disk, the removal of copy protection gives the user the option of modifying application programs to meet his or her needs.

New readers are advised to read over the rest of this page carefully in order to avoid frustration when following the softkeys or typing in the programs printed in this issue. Longtime readers should know what to do next: Make a pot of coffee, get out some blank disks and settle in for a long evening at the keyboard.

---



# Hardcore COMPUTIST

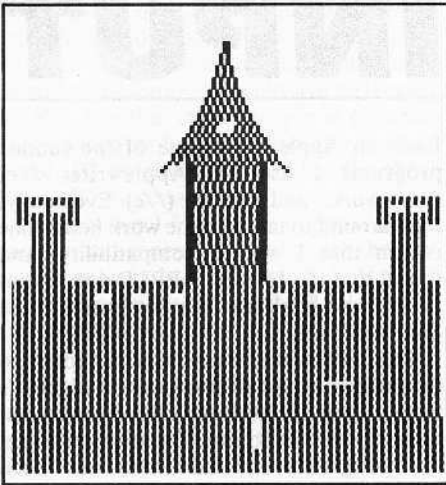
Publisher/Editor: Charles R. Haight Technical Editors: Gary Peterson, Ray Darrah

Production & Graphics: Lynn Campos-Johnson Circulation: Michelle Frank

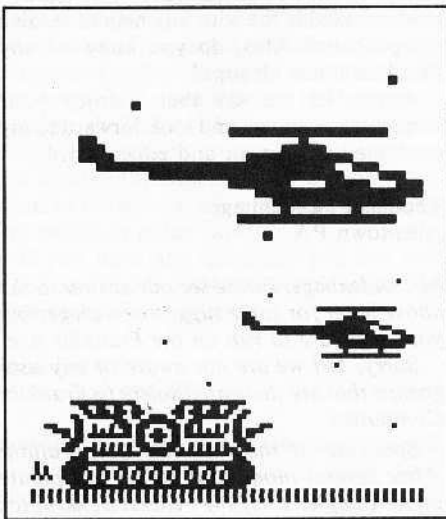
Business Manager: Valerie Robinson Advertising: (206) 581-6038 Printing: Grange Printing, Inc., Seattle, WA

Hardcore COMPUTIST is published monthly, except December, by SoftKey Publishing, 3710 100th St SW, Tacoma, WA 98499

Phone: (206) 581-6038



Page 6



Page 6

Address all advertising inquiries to Hardcore COMPUTIST, Advertising Department, PO Box 110816, Tacoma, WA 98411. Mail manuscripts or requests for Writers Guides to Hardcore COMPUTIST, PO Box 110846-K, Tacoma, WA 98411.

Return postage must accompany all manuscripts, drawings, photos, disks, or tapes if they are to be returned. Unsolicited manuscripts will be returned only if adequate return postage is included.

Entire contents copyright 1985 by SoftKey Publishing. All rights reserved. Copying done for other than personal or internal reference (without express written permission from the publisher) is prohibited.

The editorial staff assumes no liability or responsibility for the products advertised in the magazine. Any opinions expressed by the authors are not necessarily those of Hardcore COMPUTIST magazine or SoftKey Publishing.

Apple usually refers to the Apple ][ or ][ Plus Computer, and is a trademark of Apple Computers, Inc.

**SUBSCRIPTIONS:** Rates: U.S. \$25.00 for 12 issues, Canada \$34.00, Mexico \$39.00, Foreign (airmail) \$60.00, Foreign (surface mail) \$40.00. Direct inquiries to: Hardcore COMPUTIST, Subscription Department, PO Box 110846-T, Tacoma, WA 98411. Please include address label with correspondence.

**DOMESTIC DEALER RATES:** Call (206) 581-6038 for more information.

**Change Of Address:** Please allow 4 weeks for change of address to take effect. On postal form 3576 supply your new address and your most recent address label. Issues missed due to non-receipt of change of address may be acquired at the regular back issue rate.

## 9 Secret Weapon: RAMcard

If you have an Apple //e and haven't been able to RESET into the monitor for some of Hardcore COMPUTIST's softkey procedures, the auxiliary memory in your extended 80-column card may just do the trick for you. *By Ken Greenlaw.*

## 12 Deprotecting Sensible Speller for ProDOS

Another chapter in the continuing saga of Sensible Speller Softkeys. This time we uncover the secrets to the deprotection of the ProDOS version. *By Peter Rongays.*

## 15 The Controller Writer

Have you found that writing Super IOB controllers is too difficult or time consuming? Sweat no more! The Controller Writer will quickly and easily generate custom Super IOB controllers for you. You only need answer a few simple questions to get things started. *By Nick Galbreath.*

### CORE SECTION

## 19 A Fix For The Beyond Castle Wolfenstein Softkey

## 20 The Lone Catalog Arranger

Part One: The BASIC program. With this program and the program appearing in next month's issue you will be able to easily edit your disk directories. Includes these commands: file order modification, renaming, deleting and undeleting. *By Ray Darrah.*

## 25 Softkey For Sideways

Yes, it is possible to backup (and unprotect) your valuable Sideways diskette. Read this article to discover exactly how. *By Jack Burke.*

### DEPARTMENTS

## 4 INPUT

## 6 READERS' SOFTKEY & COPY EXCHANGE

**Rescue Raiders Softkey And APT's**  
*By Clay Harrell*

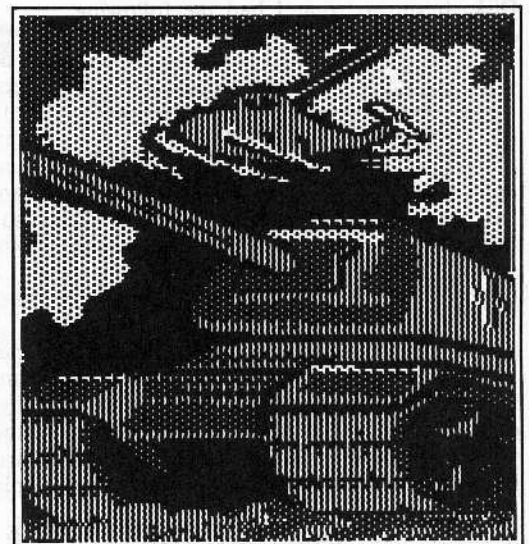
**Deprotecting Sheila**  
*By Tom Phelps*

**Deprotecting Basic Building Blocks**  
*By Doni G. Grande*

**Deprotecting Artsci Programs**  
*By Ray Darrah*

**Softkey For Crossfire**  
*By Marco Hunter*

## 14 ADVENTURE TIPS





# INPUT INPUT INPUT

## Further Beneath the Castle

Concerning Issue No. 13, page 12 on Beneath "Beyond Castle Wolfenstein": I've got a couple of extra values for your list (if you haven't got them already). They are as follows:

Disconnect alarm system 5082:60  
Reconnect alarm system 5082:A2  
Turn off alarm 1195:00/4353:00 (both)

I also use the following routine:

432A = 00 00 00 (heal yourself)

Thanks alot. this article really helped me out. Keep up the good work.

Ted Strain  
Long Beach CA

## AceWorks?

I have just completed reading and following your guidelines for "Using ProDOS On A Franklin Ace." This conversion worked GREAT on the ProDOS User's Disk. I have also used this on an Appleworks Start-up Disk but have not yet been able to get the disk to work properly. Here is a description of how it went:

When I attempted to boot a ProDOS disk without any alterations, it would continually hang. After I made the alterations as given, the ProDOS User's Disk would boot well. The problems arose in the Appleworks Start-up Disk. Without the alterations it would hang just the same as the User's Disk. After the alterations it seemed to get a little further into the program, but then it displayed in inverse video, "Appleworks Requires an 80-column //e", and then it also hung. At this point, I also attempted to use my 80-column card in slot 0 but had no success.

I then noticed that something strange occurred upon booting the ProDOS User's Disk. It showed that I had the following card/slot assignments:

Slot	Display	Actual cards
0	Language	Empty
1	Serial	Parallel Interface
2	Serial	Serial Interface
3	Serial	80-column card
4	Used	CP/M
5	Empty	Empty
6	Disk Controller	Disk Controller
7	Empty	Empty

Slot 1 is the actual location of my dual interface card and is assigned as the parallel interface. Slot 2 is the "phantom location" of my serial interface. Any help or ideas that you have would be greatly appreciated. I have spent many, many hours reading the

sector editor of the program trying to find the problem, but now I am out of ideas.

Edward J. Fitzpatrick  
Pueblo CO

*Mr. Fitzpatrick: One of Appleworks' stated hardware requirements is an 80-column card in the Apple //e's auxiliary slot. As you have discovered, Appleworks is checking for the presence of the card. Without it the program will refuse to run.*

*The usual way to check for the presence of an Apple //e is to check for the //e's "signature byte" of \$06 at \$FBB3 in the monitor ROM. On a Franklin Ace 1000 this byte is a \$EA. You may want to search the Appleworks disk for any code that is accessing \$FBB3. Appleworks is probably also checking for the presence of the 80-column card by trying to store some bytes in the 80-column RAM and then testing to see if the bytes can be read back correctly.*

*For a good example of the type of code that is used to perform this check, see the article in the February 1983 issue of CALL A.P.P.L.E. entitled, "Revvng Up the Rev e", authored by Cliff Howard and Art Schumer.*

*Of course, if you can successfully remove the checks for the auxiliary 80-column card you will also be faced with the task of modifying Appleworks so that it utilizes your particular 80-column card. Remember also that when Appleworks is run on a computer with 64K of RAM, it leaves very little free memory. Therefore, you would probably want to come up with a patch that would allow the program to utilize a RAM card in one of the standard peripheral slots.*

*Well, that just about exhausts our knowledge of Appleworks. Perhaps some of our readers will have further suggestions.*

## Aced Again

Thank you for a very enjoyable and instructive magazine. I am, at best, a novice when it comes to the software aspects of using a computer. Using BASIC to write my own programs is one thing, but trying to understand what's happening using assembly and/or machine language leaves me somewhat baffled. You should be happy to know that reading and using your magazine is helping, although it is going to be a very slow process.

Now, to my problem: A year ago I purchased a Franklin 1200 OMS, based on an economic comparison, a strong assurance by the dealer that it was very compatible with the APPLE, and a strong recommendation from a friend. Admittedly, I have been very pleased with my system. However, at work

I use an Apple //e. Some of the canned programs I use are Applewriter //e, Appleworks, and PFS File //e. Eventually, I got around to taking some work home (one reason that I wanted compatibility) and found that the folks at APPLE don't want people who don't use their computers to use their software.

The software obviously checks to determine the type of computer on which it's being run and, if it doesn't like what it sees, it simply refuses to run. My attempts to get the programs to run have not been successful, even with some of the hints I was able to get out of Hardcore COMPUTIST. Can your staff or any of your more learned readers provide me with any help to resolve my problem? Also, do you know of any Franklin Users Groups?

Again, let me say that I enjoy your magazine very much and look forward to my continued enjoyment and education.

Thomas F. Oldenhage  
Allentown PA

*Mr. Oldenhage: Please see our answer to the above letter for some suggestions on getting your software to run on the Franklin Ace.*

*Sorry, but we are not aware of any user groups that are dedicated solely to Franklin Computers.*

*Speaking of the Franklin Corporation: After several months of trying to operate under Chapter 11 of the Federal Bankruptcy Laws, they are liquidating their assets and going completely out of business. This was reported in the December 31st, 1984 issue of Infoworld. The liquidation will take about one year and the company will continue to honor its warranties during this period.*

*Franklin is currently selling all of its computers and components (bare, partially and fully assembled motherboards and peripheral cards) at very low prices to individuals and dealers. For more details on this, see their ads in the November and December issues of the Computer Shopper or call Franklin at 609-488-0600.*

## Rana Disk Drive Modification

Here is a hardware modification note that I haven't seen published in Hardcore COMPUTIST yet. It is for people who have Rana Elite One's and want to be able to write on the backside of their disks without having to cut a notch. To do the modification you will need to get a cheap 16-pin socket from Radio Shack for the actual switch soldering so that the drive will be intact for any warranty or repair work.

What you want to do is disable one pin on a single chip on the drive's card. The



correct pin in question depends on whether you have the long board (which extends the length of the drive) or the short board (only about half as long). The chip will be labeled 74LS26, in either case. The pin in question will be pin #6 for both situations (for the long board, IC number U11 and for the short board, IC number U9).

Now get two 3-foot pieces of heavy gauge single-strand copper wire and attach them to a small single-pole on/off switch. Solder one of the wires to lead #6 of the Radio Shack socket which you have previously bent 90 degrees to the rest of the pins. Strip a very small length of insulation off the other wire and insert it into the Rana #6 lead socket which had originally held the IC. Make sure that the two exposed ends never contact each other or you will defeat the purpose of the switch.

Then insert the modified socket into the Rana socket and insert the IC (make sure the orientation of the chip is the same) into your customized socket. In one position the drives are completely normal while in the second position, the Rana's LED write protect notch switch is disabled. This modification does not disable the touch switch on the front panel of the drive at all; it still functions with the switch in either position.

If you have any questions you can call Rana at 818-709-5484 and speak to one of their technicians directly.

John Roukous  
North Andover MA

*Sounds OK. But Readers, please don't call us if this modification should cause your drives to run backwards or not at all*

### An Incompatible Compatible

**Y**ou seem to have some pretty intelligent people among your readers and staff and I would like to ask them some questions about the Apple.

1) When some copy cards indicate that a RAM card is necessary, they seem to be referring to the 48K machine. What does one do when he has a //e model?

2) I have a Taiwan made Apple compatible. It is an excellent machine with numeric keyboards, lots of editing facilities and one-key entry of most BASIC commands. It also works with printer, disk drive and RS232 cards. What puzzles me is that it refuses to work with a 128K RAM card and a 6809 card which both work OK on genuine Apples. Any suggestions?

3) Can one use a Z-80B microprocessor on a Z-80 card, and would this provide any advantage?

4) What compilers do you recommend for Applesoft and MBASIC which can also handle numbers with decimals? As I understand it, some compilers only handle integer numbers.

Your magazine is most enjoyable and has helped me to gain a better understanding of computers in general. A friend of mine who has an IBM-PC was quite envious when he saw Hardcore COMPUTIST and he asked if there was a corresponding magazine for the IBM. Do you know of any?

S. Vouyoucalos  
Athens, Greece

*Mr. Vouyoucalos: Thank you for the compliments. We will try to answer your questions as best we can. Here goes:*

1) *The //e has a 64K of RAM built-in, so no extra memory cards should be needed to use any of the copy cards that are advertised as being Apple //e compatible.*

2) *The cards you own are probably pulling the INH line low in order to disable the motherboard ROMs. Unfortunately, due to the improper circuitboard design of some Apple clones, pulling this line low has the effect of disabling all I/O decoding, without which the computer will not function.*

3) *The Z80-B is faster than a standard microprocessor because it runs at a higher clock speed. Because this higher clock speed would not be available on a board set-up for a standard Z80, merely putting in the Z80-B would do you no good.*

4) *There are several good Applesoft compilers available such as Microsoft's TASC Compiler, the Einstein Compiler and Hayden's Applesoft Compiler. All of these compilers support real number variables, but only TASC supports both integer and real variables. Because TASC has no licencing requirement (other than a message in the program and its documentation) for commercial use, we prefer this compiler over its competitors.*

*As for MBASIC compilers, we really have no experience from which to make a recommendation, so we had better pass on this one.*

*As far as we know, there are no IBM-PC related publications that print material similar to that of Hardcore COMPUTIST.*

### BDR Enhancement

**I** enjoyed the Batman Decoder Ring by Ray Darrah and have enhanced it by adding a two-to-ten letter list of words plus a common-usage list which is menu-selectable in place of the QUIT function. Should any readers desire this program (it's far too long to list and format properly) have them send me a disk and a couple of quarters for postage and I'll return to them the "Improved" Batman Decoder Ring. (Incidentally, I believe line 740 should end with the "THEN 730". Am I right?)

I like Ray's disk access routine. If you're very careful you can edit the messages with a text editor/word processor. All kinds of possibilities exist for modifying BDR for

longer text, etc.

Don Soul  
Digitech  
8690 Aero Drive, Suite M-151  
San Diego CA 92123

### An Educator in Need of Help

**I** am in charge of the implementation of microcomputers in my school district and am currently have problems with backing-up three of our programs. The programs have multiple disks and two of them have no backup policy. One is Borg Warner's College Entrance Examination Preparation which I have tried, with no success, to copy. Last year, disk B broke down and I lost the records of 65 students. The disk was replaced for a modest fee but it was about five weeks before I received the replacement. With students preparing for the SAT's, that was devastating. They have no backup policy; you pay full price all the way. The second program is Comprehension Power by Milliken. We own several of these programs and currently I have 5 crashed disks on my desk. The last program is Borg Warner's Critical Reading. Again, I have several of these programs and three of the disks sitting on my desk, unusable.

I would really appreciate knowing if anyone has been successful in making a backup copy of these programs, the copier used and the parameter changes which are necessary. Perhaps putting these on your Most Wanted List will provide me with a solution. Thanks for your cooperation.

Hardcore COMPUTIST has really helped toward my utilization and understanding of copying programs. Now I need to become proficient in assembly language so I can make better use of the softkeys. Keep up the good work and keep it coming.

David J. Ward  
South Williamsport PA

*Mr. Ward: At this time we do not have any information on backing-up the programs you mention, but we have placed them on our Most Wanted List.*

Please address letters to: Hardcore COMPUTIST, Editorial Dept., PO Box 110846-K, Tacoma, WA 98411. Include your name, address and phone. Correspondence published in the INPUT section may be edited for clarity and space requirements.



# READERS' SOFTKEY & COPY EXCHANGE

## Rescue Raiders Softkey And APT's

By Clay Harrell

Rescue Raiders Ver. 1.2  
Sir-Tech Software  
6 Main Street  
Ogdensburg, NY 13369  
\$34.95

### Requirements:

Apple II, III, //e, //c  
64K RAM minimum  
At least one DOS 3.3 disk drive  
COPYA program  
A sector editor  
A blank disk

**R**escue Raiders, a recent release from Sir-Tech Software, is arguably one of the best arcade games ever created for the Apple. It might be described as a hybrid version of Choplifter and Defender with elements of Strategic Simulations' Battle for Normandy thrown in for good measure. The graphics are superb and, to me, the game is very addictive.

Rescue Raiders places you in command of air and ground forces on D-Day, June 6, 1944. At the center of the action, you control and view helicopters, ground vehicles, soldiers and missiles as you try to gain ground from the enemy forces. To win at Rescue Raiders requires both good strategy and arcade dexterity. There are eight battles, and each becomes increasingly more difficult and time consuming than the previous battle.

In addition to the game's excellence, the copy-protection is also top-notch. The protection is *meant* to be devious, which makes you want to have a backup that much more. Although you can make a backup that will run using COPYA with no modifications, a "Bad Drive Speed" message will appear and the game won't advance beyond level one. Using Locksmith or another bit copier will result in a copy that does not run at all.

Another poke-in-the-ribs from the authors of the game is a message that is stored on the text page. The message would normally go unseen because the game is done entirely in hi-res graphics. But if you are snooping around in any depth, the message becomes obvious: *"Modifying the hi-score will void your warranty and Santa won't visit you any more. Things aren't(sic) always as simple as they seem"*. This more-or-less reflects the results of a COPYA vs. Locksmith backup (notice the misspelling of "aren't" in the message!). This message can also be found on track \$0, sector \$F. Beyond the mystery message that Sir-Tech provides, the actual protection code is a *bear* to find.

When deprotecting a program it is best to first get the program into normal DOS 3.3

format and try to run it. Then, when the program doesn't work at some point (i.e. the program's DOS cannot read the normal DOS format), trace back to find the problem.

Unfortunately, Rescue Raiders is already in normal DOS format and there is no hint of trouble when running a COPYA copy except for the "Bad Drive Speed" message and the inability to advance beyond level one. So, where do you start? The problem code could be practically anywhere in the boot and could be read in, executed, and destroyed within a few milliseconds.

The answer to this \$64 question is, "Start at the bottom," or boot-code trace. Alternatively, you could find the routine that prints the hi-res message "Bad Drive Speed" and work backwards, but that too is not an easy matter. To make a long story short, I have done all of the leg work for you, and then some. Beyond the general approach already mentioned, I don't really the time to do a proper write-up on the subject, so I will instead provide a cookbook procedure which will have to suffice.

### Copying Rescue Raiders

1) Boot normal DOS 3.3 and run COPYA from your DOS 3.3 System Master by typing

#### RUN COPYA

2) Copy the original Rescue Raiders disk to a blank disk.

3) Run your sector editor and make the following changes to your COPYAed copy

Track	Sector	Byte	From	To
\$1E	\$00	\$50	\$87	\$C7
\$1E	\$00	\$67	\$0E	\$C7

Write the sector back out to the COPYAed copy. These sector edits will defeat the "Bad Drive Speed" error message.

4) To defeat the level problem, the following sector edits are necessary

Track	Sector	Byte	From	To
\$0E	\$0B	\$51	\$A2	\$18
\$0E	\$0B	\$52	\$03	\$A9
\$0E	\$0B	\$53	\$8D	\$80
\$0E	\$0B	\$54	\$A6	\$8D
\$0E	\$0B	\$55	\$60	\$AC
\$0E	\$0B	\$56	\$1D	\$60
\$18	\$02	\$57	\$A7	\$F0
\$18	\$02	\$58	\$60	\$43

5) Write the sector back out to your COPYAed copy to defeat the protection from the original Rescue Raiders disk.

### Free APT's

The authors of Rescue Raiders were no dummies and realized that to advance easily to each level to test it would require a "cheat mode". Fortunately, for us, this mode remained in the version of the game that was released to the public. I wonder if this was intentional or an oversight on the part of the

authors.

I discovered the "cheat mode" while doing some disk snooping during the deprotection process. Some text at track \$0F, sector \$9, byte \$A8 to \$AC turned out to be the password that evokes the special mode. The password on my copy is "POPPY", (spelled backwards at bytes \$A8 to \$AC). I could now easily and quickly test my deprotection.

After the play of the game has started and you have control over the helicopter, type "POPPY" to activate the "cheat mode".

Here are the Cheat mode commands:

/ = Free Helicopters up to a maximum of 255  
J = Instantly moves you to farthest left of playing field  
K = Instantly moves you to dead middle of playing field  
L = Instantly moves you to farthest right of playing field  
RETURN = Makes you bullet proof (but not missile proof)  
-# = Pressing the "-" key followed by a number 1 to 9 will end your current battle and put you at that desired battle (level). Number 9 ends the game.

The game is still difficult even with these commands. You can never be missile proof and there are no commands at all to manipulate enemy troops. To change the password, you may sector edit track \$0F, sector \$9, bytes \$A8 to \$AC.

If the password "POPPY" does not work on your copy of Rescue Raiders, you may want to check track \$0F sector \$9 to see if Sir-Tech changed it (read backwards!). And, good luck! Believe me, you'll need it to beat Rescue Raiders.

### Deprotecting Sheila

By Tom Phelps

#### Sheila

H.A.L. Labs  
4074 Midland Rd., Ste. 23  
Riverside, CA 92505  
(714) 359-8480  
\$25.00

#### Requirements:

48K Apple or compatible  
Sheila original disk  
One blank disk

The Inspector or similar program

*It's only been an hour  
Since he locked her in the tower.  
The time has come  
He must be undone  
By the morning*

**A**nd so begins Sheila, an excellent action - adventure - arcade type game of the Castle Wolfenstein ilk. I suspect that its limited distribution is the reason that it hasn't enjoyed the same kind of popularity its cousin Wolfenstein has.



At any rate, this one gives the bit copiers fits! Just look at any parameter listing and it's not listed. These lists always say that if a program isn't listed that it may copy without any parameter changes. Unfortunately, this isn't the case with Copy [( Plus or EDD. Who likes copies that have to be nibble copied, anyway? Let's tackle the task of deprotecting this offering from Brian Fitzgerald of Softline, St. Game fame and his company, H.A.L. Labs!

Since this game uses the disk during play, a single Binary file is not a likely candidate. Therefore, we need to copy the entire disk so that we obtain the main program, each of the five mazes and the hi-res castle at the beginning.

Let's approach this thing intelligently so you will be able to crack a program with similar protection later (following an obscure list does not help one's cracking skills much.) First, listen to the boot of the original disk. Next, listen to one of a normal disk. Sound similar? They are indeed much alike except for a complex DOS track check (a nibble count, perhaps?) on the Sheila disk. This tells us, the experienced computists that we are, that our old friend Super IOB might be helpful.

Scanning the disk with the Inspector's nibble read, we find that only tracks \$00-\$16 have information on them. We also learn that on tracks \$01-\$16 the last byte of both the address and data markers have been changed.

Incorporating all of this information into a Super IOB controller, we come up with the one printed at the end of this article. The controller performs a few sector edits that change the Sheila DOS to reflect its now unprotected format. The bytes changed are:

Track	Sector	Byte	From	To
\$00	\$02	\$5D	\$DA	\$AD
\$00	\$02	\$FC	\$DA	\$AD
\$00	\$03	\$6A	\$AD	\$96

The sector edits performed by the Super IOB controller which disable the psuedo nibble counter are:

Track	Sector	Byte	From	To
\$01	\$03	\$04	\$20	\$EA
\$01	\$03	\$05	\$2C	\$EA
\$01	\$03	\$06	\$83	\$EA

The Super IOB controller makes a COPYAable version of Sheila. Have fun and happy cracking!

### The Controller

```

1000 REMSHEILA CONTROLLER
1010 TK = 0 : ST = 0 : LT = 35 : CD = WR
1020 T1 = TK : GOSUB 490
1025 IF TK <> 0 THEN RESTORE : GOSUB
      190 : GOSUB 210
1030 GOSUB 430 : GOSUB 100 : ST = ST + 1
      : IF ST < DOS THEN 1030
  
```

```

1040 IF BF THEN 1060
1050 ST = 0 : TK = TK + 1 : IF TK < LT
      THEN 1025
1060 GOSUB 310 : GOSUB 230 : GOSUB 490
      : TK = T1 : ST = 0
1070 GOSUB 430 : GOSUB 100 : ST = ST + 1
      : IF ST < DOS THEN 1070
1080 ST = 0 : TK = TK + 1 : IF BF = 0 AND
      TK < LT THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "DONE^WITH^COPY" :
      END
5000 DATA 213 , 170 , 173
5010 DATA 213 , 170 , 218
5020 DATA 6CHANGES
5030 DATA 0 , 2 , 93 , 173 , 0 , 2 , 252
      , 173 , 0 , 3 , 106 , 150
5040 DATA 1 , 3 , 4 , 234 , 1 , 3 , 5 , 234 , 1
      , 3 , 6 , 234
  
```

### Controller Checksums

1000 - \$3568	1080 - \$1E0E
1010 - \$3266	1090 - \$6510
1020 - \$C11A	1100 - \$F73F
1025 - \$1E31	5000 - \$85BD
1030 - \$0830	5010 - \$F7AD
1040 - \$824D	5020 - \$CB8F
1050 - \$071D	5030 - \$3C35
1060 - \$68B2	5040 - \$5615
1070 - \$60B7	

### Deprotecting Basic Building Blocks

By Donl G. Grande

Basic Building Blocks  
Micro Educaton Corporation of America  
285 Riverside Ave.  
Westport, CT 06880  
(203) 222-1000  
\$79.95

Requirements:  
Apple with 48K  
A copy program which will copy a range of tracks or ignore read errors  
A sector editor  
Two blank disks

**B**asic Building Blocks (BBB) is an educational program designed to teach Applesoft and is a fairly well done package. The heart of BBB is the Basic Design Tool (BDT) which allows one to step and trace a basic program while watching the effect on the variables. BDT may be used to debug your own programs, and this alone makes BBB a very useful program package.

BBB uses an interesting protection scheme which is easily defeated. Track 34 (\$22) on disk 1 (it is a two disk package) is written with altered address (D5 AA 97 instead of D5 AA 96) and data (D5 AA AC instead of D5 AA AD) marks. However, the program only checks this in one place, by attempting to read track \$22, sector \$03, and then does nothing with the data read. To defeat this

copy protection, do the following:

- 1) Copy the original disk, ignoring any errors on track 34 (\$22).
- 2) Using a sector editor, make the following changes:

Track	Sector	Byte	From	To
\$18	\$02	\$43	\$20	\$18
\$18	\$02	\$44	\$D9	\$EA
\$18	\$02	\$45	\$03	\$EA

This change is to the file HIRESC, and clears the carry flag instead of calling RWTS. If you look at this area of the original code, notice that the address and data marks are patched, DOS is called to read T\$22,\$\$03 and, if the read is successful, the address and data marks are restored to their correct values.

- 3) Copy Disk 2 normally, and you now have the complete set.

### Deprotecting Artsci Programs

By Ray Darrah

Artsci, Inc.  
5547 Satsuma Avenue  
N. Hollywood, CA 91610  
(213) 985-2922

Requirements:  
Super IOB v1.2  
An Artsci Program

**A**fter reviewing a number of Artsci programs recently, I noticed that they all had a common protection scheme: normal DOS format on most of the disk and a changed end of data marker on the sectors that contain the protected program. I firmly believe that utility programs (such as ACEcalc, Magic Words and Magic Mailer) should, more than any others, be unprotected. Therefore, I decided to undergo the task of deprotecting these villains.

### The Procedure

The following is an account of my method for discovering the technique of deprotecting these programs. If the machine language of it all doesn't interest you, then you may skip to the paragraph labeled, "Step By Step."

Since the Artsci disk could be CATALOGed under normal DOS I decided to try to copy the diskette with Locksmith 16-sector Fast Copy. The appearance of an inverse D indicated that the DATA field was messed up on certain tracks of the disk. Had an A appeared, that would have indicated an incorrect address field.

Unfortunately, Locksmith Fast Copy doesn't provide the specifics of the incorrect field. I then had to search through many bytes, on a track which I knew to have a bad



data field, with my nibble editor (The Linguist from CIA) until I found the alteration. The data field epilog (ending) byte pattern had been changed from the normal DEAA to the strange D5AA. Armed with Super IOB v1.2 and the standard controller, I changed it so that it would ignore the first byte of the end of data pattern. This was accomplished by placing a POKE 47415,0 in line 1010.

When I attempted to boot the copy, however, it wouldn't. Obviously, during the bootup procedure, a program changes the end of data pattern that the normal DOS on the disk looks for. Therefore, I needed to get out my sector editor (DiskEdit from The Best of Hardcore Computing) and make a change to the DOS on the disk that would make it ignore the first byte of the end of data marker.

This new copy of the original booted and worked perfectly! I then updated my Super IOB controller to perform this sector modification (Track 0, Sector 3, Byte \$37, From \$0A, To \$00) and, Voila! A controller fit to deprotect any Artsci program.

### Step-By-Step

The following is a by-the-hand procedure for deprotecting just about any Artsci program:

- 1) Load your copy of Super IOB v1.2

#### LOAD SUPER IOB 1.2

- 2) Type the following controller

```
1000 REMACECALC CONTROLLER
1010 TK=0:ST=0:LT=35:CD=WR:
    POKE 47415,0
1020 RESTORE:T1=TK:GOSUB 490
1030 GOSUB 430:GOSUB 100:ST=ST+1
    :IF ST<DOS THEN 1030
1040 IF BF THEN 1060
1050 ST=0:TK=TK+1:IF TK<LT
    THEN 1030
1060 GOSUB 310:GOSUB 490:TK=T1
    :ST=0
1070 GOSUB 430:GOSUB 100:ST=ST+1
    :IF ST<DOS THEN 1070
1080 ST=0:TK=TK+1:IF BF=0 AND
    TK<LT THEN 1070
```

### ☆☆☆ IMPORTANT! ☆☆☆

#### Note To Subscribers:

Hardcore COMPUTIST does not send subscription renewal notification. This notice appears only on your mailing label. To renew, fill out the form on page 27.

Please note that the Post Office does not inform our subscription department when you file a change of address. You MUST send a change of address to Hardcore COMPUTIST on postal form 3576 (or use the form on page 28 of this issue). Issues missed due to non-receipt of change of address can be purchased at the current back issue rate.

```
1090 IF TK<LT THEN 1020
1100 HOME:PRINT "DONE^WITH^COPY":
    END
5000 DATA 1^CHANGES
5010 DATA 0,3,55,00
```

- 3) Run this modified Super IOB program  
RUN

4) Follow the prompts and insert your disks in their correct drive(s).

5) Put your original disk in your safe deposit box and make several hundred thousand backup copies of the the copy you just made.

Enjoy your more useful deprotected (COPYAable) utility program.

## Softkey for Crossfire

By Marco Hunter

Sierra On-Line, Inc.  
36575 Mudge Ranch Road  
Coarsegold, Ca 93614  
\$29.95

#### Requirements:

48K Apple or equivalent

COPYA

A sector editor

One initialized slave disk and one blank disk  
Joystick/keyboard version of Crossfire

**W**hen you begin to examine Crossfire, the first thing that you should note is that it is on a normal DOS 3.3 disk, and that it even has a catalog. However, the catalog does look suspicious. It does not display the file lengths or their types. It's obvious that On-Line is trying to hide something in the catalog.

To see what is really on the disk use your sector editor to read in track 11, sector F, the first track of a normal catalog. "Crossfire", the first file, is the one we are interested in. Right before "Crossfire" you will see a number of blanks (88's in hex). These are backspaces (<sup>CR</sup>H). When you catalog the disk from normal DOS, the backspaces cover over the file types and lengths with "Crossfire".

The file "Crossfire" is actually only a loader, and not the game. In front of the first 88 you will see 04. This is the file type and status. 04 is a binary file. If it were 84 (04+80) it would be a locked binary file. The two bytes before 04 tell where the track-sector list is for this file. The track-sector list identifies the sectors on which your program is stored. Mine (and hopefully yours, too) are 17 and F. Read in track 17, sector F. The first sector that "Crossfire" uses is track 17, sector E. Read this one in next. When you disassemble the sector (ZAP from Bag of Tricks is good for this), you will spot a jump to \$4000 at bytes \$23-\$25. This seems very strange because the files starts at \$A43 and doesn't reach into \$4000. To tell where the file starts, take the first two bytes of the first

sector and reverse them (i.e. 0040 is \$4000). This jump does, in fact, start the game once it has been loaded.

Now that we have the starting location, we need to know only two other things: memory used and the location of the nibble count. A stroll down memory lane reveals that \$2000-\$9500 is used. After some trial and error experimentation I discovered that the disk-check starts at \$5927. With this information we can easily save Crossfire into one file.

- 1) COPYA your Crossfire disk onto the blank disk.
- 2) With the sector editor modify

Track	Sector	Byte	From	To
17	0E	24	00	59
17	0E	25	40	FF

- 3) Boot the copied Crossfire.
- 4) When it drops into the monitor remove the call to the nibble count by typing

5927:EA EA EA

And insert a jump to the start of the program by typing

1FFD:4C 00 40

- 5) Boot up your slave disk (preferably one with a fast DOS) and type

BSAVE CROSSFIRE, A\$1FFD,L\$7504

## Who Wants A 65C02?

**H**ardcore COMPUTIST is planning to make a volume purchase of 65C02 microprocessors should a sufficient number of readers indicate that they would like to purchase one of the chips. The main advantage of the 65C02 microprocessor is an enhanced instruction set.

The 65C02 microprocessor from NCR is being used in the Apple //c but can be substituted for the 6502 in Apple //e's. Apple II and II Plus owners can also utilize the 65C02, but may have to replace two 74LS257 chips on the motherboard with faster 74F257's (74F257's will also be available for those who need them). We plan to purchase the chips from either NCR or GTE.

If you are interested in a 65C02 for your Apple or compatible, drop us a line and let us know what kind of system you have (II, II Plus, //c or Apple-compatible (please specify brand)). You will be notified if/when you may place your order. The tentative cost for a single chip is under \$10. Send your letter to:

Hardcore COMPUTIST  
PO Box 110846C  
Tacoma, WA 98411



# Secret Weapon: RAMcard

By Ken Greenlaw

## Requirements:

Apple //c or Apple //e with extended 80-column card

**W**hen Apple designed the Apple //e, they made it expandable to 128K through the use of their Extended 80-Column Card. With the release of the Apple //c they took things a step further by making the 128K of memory a standard feature. Most people use the extra memory for implementing double hi-res graphics or for extra data storage by such programs as Appleworks. A little-known fact, however, is that this memory can also be used as a tool for unlocking protected software. With the use of this memory and a little bit of

programming, no longer will //e and //c owners have to dread reading the instructions to "Hit RESET and enter the monitor" which so often appear in the pages of this magazine.

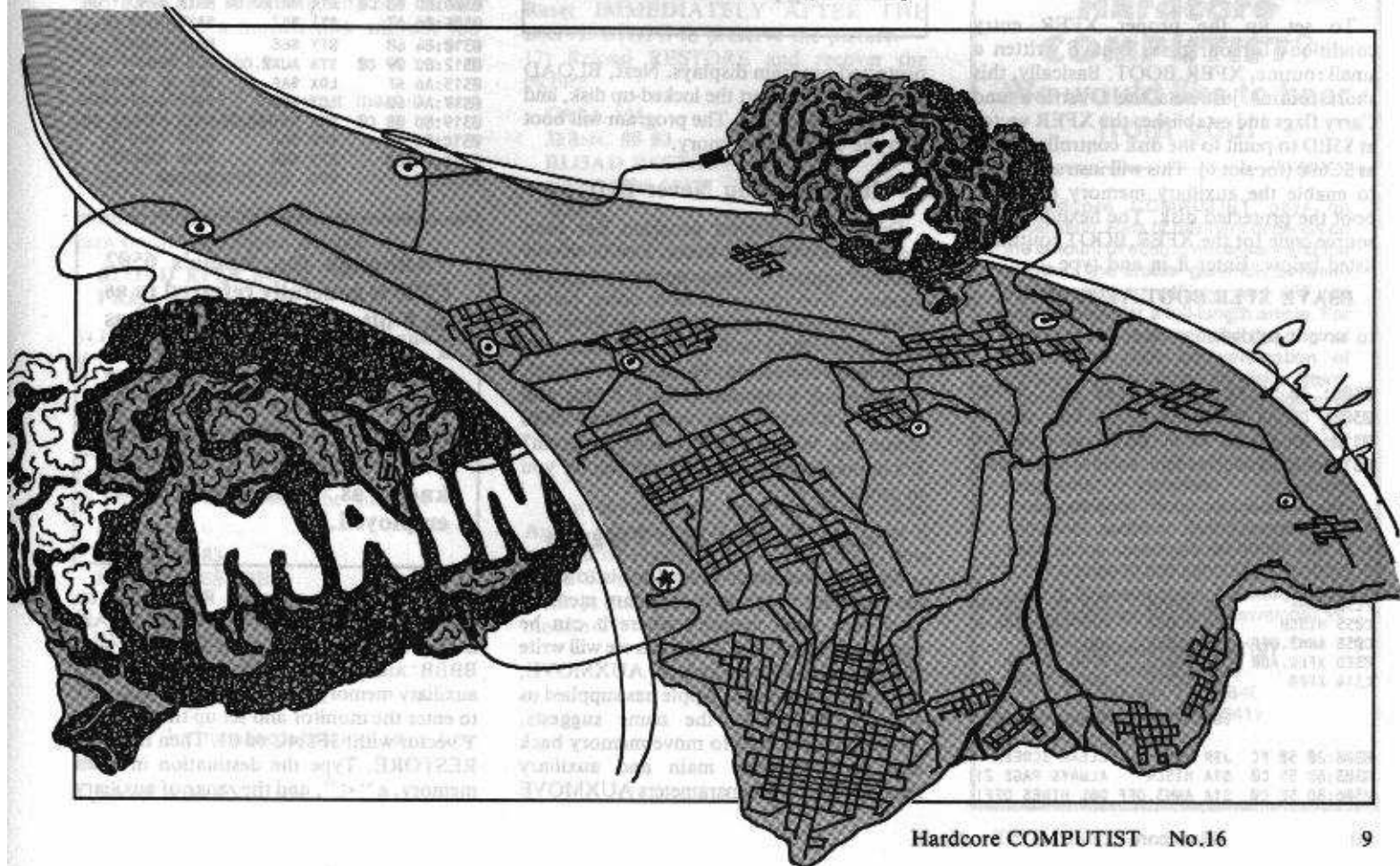
## The Theory

Before discussing how auxiliary memory can be used for deprotection purposes, lets take a look at how the Apple accesses this extra memory. Although the Apple's 6502 CPU is generally referred to as an 8-bit microprocessor, it has a 16-bit address bus which allows it to access 64K, or 65536 separate locations in memory. To be able to utilize more than 64K of memory a technique

known as "bank-switching" is employed. With this technique, extra memory can be used by mapping it into the 6502's normal 64K address range. On an Apple //e, //c or a ][+ with a 16K language card, bank-switching is used to map the additional memory into the address range normally occupied by the Apple's ROMs at \$D000-\$FFFF. This is what allows Applesoft and Integer BASICs to simultaneously reside in the Apple.

On the //c and //es with extended 80-column cards, the entire 64K address space, and not just \$D000-\$FFFF, is bank-switchable. To allow programmers to access

*Continued on next page*





the extended memory, Apple has provided a vector at \$C314 to a routine called XFER. When XFER is entered, via a JMP rather than a JSR, the state of the Carry (C bit) and overflow (V bit) are used to indicate whether main or auxiliary memory is to be switched in and whether to use the main or auxiliary zero page and stack. If the Carry bit is clear (C=0), main memory will be used; otherwise, the auxiliary memory is enabled. A cleared Overflow flag (V=0) instructs XFER to use the main zero page and stack, while a set Overflow flag (V=1) indicates that the auxiliary zero page and stack are to be used. The XFER routine also must have a vector set up at \$3ED-\$3EE as a pointer to the location of the code which will be executed upon completion of the XFER routine.

This XFER routine comes in very handy for deprotection purposes. It can be used to enable the auxiliary memory before booting up with a protected disk. Once the protected program is happily executing in the auxiliary memory, control can be returned to main memory with a Control-RESET, thus leaving the protected program "captive" in the auxiliary memory. Another ROM routine, AUXMOVE at \$C311, can then be utilized to move the captured program back to main memory where it can be examined or saved to a normal disk. This technique is applicable to just about any program that does not utilize the extended memory for its own purposes.

### Using XFER

To set up the proper XFER entry conditions for our task, I have written a small routine, XFER.BOOT. Basically, this short routine just sets the Overflow and Carry flags and establishes the XFER vector at \$3ED to point to the disk controller ROM at \$C600 (for slot 6). This will instruct XFER to enable the auxiliary memory and then boot the protected disk. The hexdump and source code for the XFER.BOOT routine is listed below. Enter it in and type

**BSAVE XFER.BOOT,AS300,LS1A**

to save it to disk.

```
0300: 20 58 FC 8D 55 C0 8D 5E   $FF02
0308: C0 A9 00 8D ED 03 A9 C6   $AC5A
0310: 8D EE 03 18 69 80 38 4C   $ADDC
0318: 14 C3                          $C073
```

\* XFER.BOOT  
\* BY KEN GREENLAW

```
FC58 HOME .EQ $FC58 CLR TXT SCREEN
C055 HISC .EQ $C055 DISPLAY PAGE 2
C05E ANN3.OFF .EQ $C05E DBL HIRES
03ED XFER.ADR .EQ $03ED XFER VECTOR
C314 XFER .EQ $C314 ENTER XFER
```

.OR \$0300 START = \$300

```
0300:20 58 FC JSR HOME CLEAR SCREEN
0303:8D 55 C0 STA HISC ALWAYS PAGE 2!
0306:8D 5E C0 STA ANN3.OFF DBL HIRES OFF!
```

```
0309:A9 00 LDA #500 LSB OF $C600
030B:8D ED 03 STA XFER.ADR STORE LSB
030E:A9 C6 LDA #500 MSB OF $C600
0310:9D EE 03 STA XFER.ADR+1 STORE MSB
0313:18 CLC
0314:69 80 ADC #500 FORCE OVERFLOW
0316:38 SEC USE AUXILIARY
0317:4C 14 C3 JMP XFER GO DO IT
```

Note that XFER.BOOT cannot be used on programs that use the auxiliary memory. These programs will know better than to load into the auxiliary 64K.

### Capturing a Program

To use XFER.BOOT, boot up DOS 3.3 and then turn on the 80-column display with PR#3. You must first see any text the

**"When Apple designed the Apple //e, they made it expandable to 128K...and with the release of the Apple //c they took things a step further by making the 128K of memory a standard feature. Most people use the extra memory for implementing double hi-res graphics or for extra data storage...A little-known fact, however, is that this memory can also be used as a tool for unlocking protected software."**

protected program displays. Next, BLOAD XFER.BOOT, insert the locked-up disk, and then type CALL 768. The program will boot into the auxiliary memory.

### It's Like You Never Left!

Once the protected program has been loaded, press Control-Reset. You have now returned control of the computer to the main memory. Everything will be in order, just as it was before you pressed Control-Reset the first time. DOS will still be intact and completely connected. To confirm this fact, try inserting a normal DOS 3.3 disk and typing CATALOG or some other DOS command. Where is the protected program? It's in the auxiliary memory waiting for you to come get it.

### Raiders Of The Lost Program

Now all that needs to be done is to move the captured program in auxiliary memory back into main memory where it can be examined, saved, etc. To do this we will write another small routine and use AUXMOVE, the second subroutine Apple has supplied us with in ROM. As the name suggests, AUXMOVE is used to move memory back and forth between main and auxiliary memory banks. The parameters AUXMOVE

needs are: the beginning and end addresses of the memory to move from, the start address of the memory to move to and direction to move (auxiliary to main or main to auxiliary). Upon entry, the status of the Carry bit indicates to AUXMOVE the direction of transfer (C=1 for main to auxiliary, C=0 for auxiliary to main).

The RESTORE routine listed below sets up AUXMOVE to transfer the captured program and Applesoft program pointers back to main memory. RESTORE will be called through the Control-Y vector at \$3F8. The hexdump and source code for RESTORE are also listed below, so type in the hexdump, and save it to disk with a

**BSAVE RESTORE,AS300,LS21**

```
0300: 18 20 11 C3 8D 09 C0 A6   $44AB
0308: 67 A4 68 8D 08 C0 86 67   $9653
0310: 84 68 8D 09 C0 A6 AF A4   $D830
0318: 80 8D 08 C0 86 AF 84 80   $6D11
0320: 60                          $ABDD
```

\* RESTORE  
\* BY KEN GREENLAW

```
C311 AUXMOVE .EQ $C311 ENTRY POINT
C009 AUX0.ON .EQ $C009 AUX PAGE 0 ON
C008 MAIN0.ON .EQ $C008 MAIN PAGE 0 ON
```

```
.OR $0300
0300:18 CLC MOVE AUX>MAIN
0301:20 11 C3 JSR AUXMOVE CALL IT
0304:8D 09 C0 STA AUX0.ON AUX MEMORY ON
0307:A6 67 LDX $67 PROGRAM START
0309:A4 68 LDY $68 FROM AUXILIARY
030B:8D 08 C0 STA MAIN0.ON MAIN MEMORY ON
030E:86 67 STX $67 SAVE THEM IN
0310:84 68 STY $68 MAIN MEMORY
0312:8D 09 C0 STA AUX0.ON AUX MEMORY ON
0315:A6 AF LDX $AF END OF PROGRAM
0317:A4 80 LDY $80 FROM AUXILIARY
0319:8D 08 C0 STA MAIN0.ON MAIN MEMORY ON
031C:86 AF STX $AF SAVE THEM IN
031E:84 80 STY $80 MAIN MEMORY
0320:60 RTS RETURN
```

**"Although the Apple's 6502 CPU is generally referred to as an 8-bit microprocessor, it has a 16-bit address bus which allows it to access 64K, or 65536 separate locations in memory. To...utilize more than 64K of memory a technique known as "bank-switching" is employed."**

The syntax for using RESTORE is: AAAA<BBBB.CCCC<sup>CH</sup>Y, where AAAA is the destination in main memory, and BBBB and CCCC define the range of auxiliary memory to move. Type CALL -151 to enter the monitor and set up the Control-Y vector with: 3F8:4C 00 03. Then BLOAD RESTORE. Type the destination in main memory, a "<", and the range of auxiliary

memory to move (separated by a period) then a **CTRL Y**. For example: To move everything from \$B800 to \$BFFF in auxiliary memory to \$1900 in main memory, type

**1900 < B800.BFFF CTRL Y**

### Classic Cases

As an example of how all of the above works, step-by-step instructions for deprotecting "Apple Adventure" from Apple Computer follow:

## Deprotecting Apple Adventure

**Apple Adventure**  
**Apple Computer**  
**28525 Mariani Avenue**  
**Cupertino, CA 95014**  
**\$35.00**

### Requirements:

Apple //c or //e with Extended 80-column card  
XFER.BOOT & RESTORE  
Apple Adventure  
FID or other file-by-file copier  
One blank disk

**O**f the nine files on the Apple Adventure disk, three of them are unprotected text files. One other file, CHAIN, can be found on the DOS 3.3 System Master disk. The other five are protected, but can be transferred to a normal disk without too much difficulty.

1) Boot DOS 3.3 into your machine and initialize a blank disk

**FP**  
**INIT ADVENTURE**

2) Load XFER.BOOT into the machine and turn on the 80-column display

**BLOAD XFER.BOOT**  
**PR#3**

3) Insert the Apple Adventure disk and boot it

**CALL 768**

4) When you are asked, "Would you like to continue a game previously saved on disk", press Control-Reset.

5) Load RESTORE into the machine

**CALL-151**  
**3F8:4C 00 03**  
**BLOAD RESTORE**

6) Now let's regain the first file on the disk, ADVENTURE

**800 < 800.95FF CTRL Y**  
**3D0G**

7) Insert the initialized disk and save these files

**SAVE ADVENTURE**

**BSAVE OBDW.OBJ, A\$175E, LSB90**  
**BSAVE VOCEB, A\$8A6F, LSB90**

8) Turn on the 80-column display and reload XFER.BOOT

**PR#3**  
**BLOAD XFER.BOOT**

9) Insert the Apple Adventure disk and boot it

**CALL 768**

10) When the game says, "Welcome to Adventure!! Would you like instructions?", press Control-Reset.

11) Reload RESTORE into the machine

**CALL-151**  
**3F8:4C 00 03**  
**BLOAD RESTORE**

12) Move this file back with

**800 < 800.95FF CTRL Y**  
**3D0G**

13) Save this file (note the hidden control-character)

**SAVE ADVENTURE CTRL Q PART TWO**

14) Turn on the 80-column display and reload XFER.BOOT

**PR#3**  
**BLOAD XFER.BOOT**

15) Insert Apple Adventure and reboot

**CALL 768**

16) When the logo appears, press Control-Reset IMMEDIATELY AFTER THE DRIVE STOPS to preserve the picture.

17) Reload RESTORE and recover the picture

**CALL-151**  
**3F8:4C 00 03**  
**BLOAD RESTORE**  
**2000 < 2000.3FFF CTRL Y**  
**3D0G**

18) Save the logo on the initialized disk

**BSAVE LOGO, A\$2000, L\$2000**

19) Use FID or any other file-by-file copier to copy ARB6, LOCF, and OBDS from the Apple Adventure disk to the initialized disk.

20) Copy CHAIN from the DOS 3.3 System Master onto the initialized disk.

### Closing Comments

You should now have an unprotected Apple Adventure. If you do not have a method of resetting into the monitor, this could be the answer for you on all those softkeys that instruct you to, "Reset into the monitor."



**Hardcore COMPUTIST** welcomes articles of interest to users of the Apple II (or compatible) computers and would like to publish well-written material including:

- Softkeys
- Hardware Modifications
- Advanced Playing Techniques
- DOS modifications
- Utilities
- Product reviews
- Adventure Tips
- Original programs
- Do-it-yourself hardware projects
- General interest articles
- Bit-Copy Parameters

Send your submission on a DOS 3.3 disk using an Apple (or compatible) editing program. Enclose a double-spaced hardcopy manuscript (typewritten or computer printed). Submissions will be returned only if adequate packaging is enclosed.

**Have you written an ARTICLE or PROGRAM you'd like to see published in Hardcore COMPUTIST? We would like to hear from you!**

**Hardcore COMPUTIST** pays on publication. Rate of payment depends on the amount of editing necessary and the length of the article- generally between \$10 for a short softkey, and \$50 per typeset page for a full-length article. For a higher pay rate, enclose the original commercial disk for verification of softkeys. We guarantee the disk's return.

Softkey Publishing buys all rights as well as one-time reprint rights (for upcoming BEST OF Hardcore) on general articles, and exclusive rights on programs. However, alternate arrangements may be made with individual authors, depending on the merit of the contribution.

For a copy of our WRITER'S GUIDE, send a business-sized (20-cent) SASE (self-addressed, stamped envelope) to:

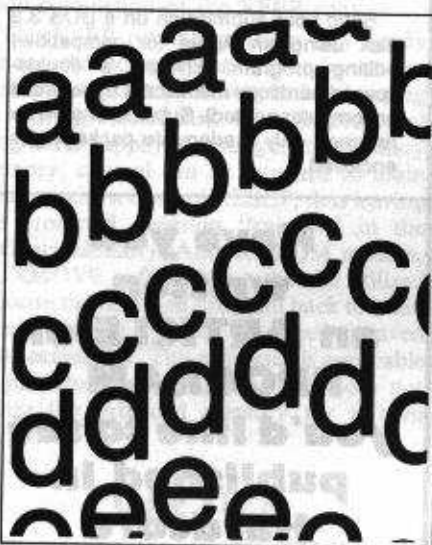
**Hardcore COMPUTIST**  
Writer's Guide  
PO Box 110846-K  
Tacoma, WA 98411



# Deprotecting Sensible Speller for ProDOS

Although softkeys for Sensible Speller IV were presented in both Hardcore COMPUTIST No.'s 9 and 11, neither will successfully backup the new version, Sensible Speller for ProDOS. Following the procedure below should allow owners of the new version to make as many backups as needed.

By Peter Rongays



Sensible Speller for ProDOS  
Sensible Software, Inc  
24811 Seneca  
Oak Park, MI 48237  
\$125.00

Requirements:  
64K Apple ][ Plus or equivalent  
Sector Editor  
Disk Search Utility  
One blank disk

Recently, I purchased Sensible Speller for ProDOS as an upgrade from Sensible Speller IV so that I could use this excellent spelling checker on files produced by Appleworks. One of the first things I did after receiving the upgrade was to try to make a backup of it using the Sensible Speller softkey procedure that was described in Hardcore COMPUTIST No.'s. 9 and 11. I was rather disappointed to learn that the procedure would not work on my new ProDOS version. None of the bit copiers I own would touch it either, so I decided to examine the disk a bit in the hope that I could develop a softkey for it. After a couple of hours of disk snooping, I was quite pleased to come up with a

procedure that would produce a working backup. This article describes my quest for the softkey procedure. If you are not interested in the details, you may want to skip ahead to the step-by-step instructions for making the backup.

## The Quest

The first thing I did in my investigation of SS-ProDOS's protection scheme was to remove the cover from my drive and watch what tracks were accessed as the original disk booted. From this little exercise I discovered that that no half, quarter or tracks beyond \$22 were being used.

Next, I thought I would try to use COPYA on SS-ProDOS, on the off-chance that it would produce a working backup. After I booted up the copy, for a time, everything seemed to be working perfectly; the ProDOS version number was displayed followed by the appearance of the hi-res Sensible Software logo while some disk access was being performed. The boot drive then shut down and the second drive, although there was no disk in it, came on momentarily. I thought I was home free until the boot drive suddenly came back on and did not shut itself off. "Oh, no! Not the dreaded nibble count," I said to myself. At this time, the drive head was positioned over track \$0. To my surprise, when I removed the copy and inserted the original SS-ProDOS disk, I was rewarded by the appearance of the Sensible Speller menu. Everything functioned perfectly once the nibble count routine had been satisfied.

When booting up the copy of SS-ProDOS, I noticed that before the nibble count of track 0 was performed, the second drive in my system would momentarily come on. I seemed to recall that while reading through the ProDOS Technical Reference Manual, I had seen a reference to a Machine Language Interface (MLI) call that would check the system to see what volumes were currently mounted. Hopefully, SS-ProDOS would be using this MLI call and I could use it to help track down the location (in memory) of the nibble count.

I opened my ProDOS Reference Manual

and, sure enough, on page 55 there was the description of an MLI call that would determine the names of all ProDOS volumes currently on-line or determine the name of a disk in a specific slot and drive. This MLI routine is called ON\_LINE and is call number \$C5. In case you don't know how the MLI is utilized, allow me to digress for a moment with the following discussion.

## The ProDOS Machine Language Interface

One of the "features" of ProDOS is the presence of consistent, interruptable and documented routines which can be called through machine language for directly accessing a disk. This set of routines, which can be accessed through the Machine Language Interface, provides programmers an easy means of handling most disk-related chores. To utilize one of the MLI routines you first set up a JSR to the MLI at \$BF00. The MLI takes the first byte beyond the JSR MLI instruction as the call number and uses the next 2 bytes as a pointer to a parameter list for that particular call. For instance, to use the ON\_LINE routine (call number \$C5) the following code would be needed:

```
MLI .EQ $BF00
ON_LINE .EQ $C5
PARMLIST .EQ $2000

20 00 BF      JSR $BF00
C5           .DA #ON_LINE
00 20       .DA PARMLIST
```

For this example, the MLI would expect to find ON\_LINE's 4 byte parameter list at \$2000.

To see if SS-ProDOS was using the MLI's ON\_LINE routine, I got out my disk search utility and set it up to search the disk for a hexadecimal sequence of 20 00 BF C5. I was surprised when this sequence was found in six different sectors on the disk. Since I wanted to know which of these calls to ON\_LINE was being executed just prior to the nibble count routine, I got out my sector editor and one by one changed the JSR opcodes (the 20

in the sequence 20 00 BF C5) to BRK opcodes (00's).

The first three BRK's I inserted didn't have any affect on the program but, after inserting a BRK into the fourth place I found the call to ON.LINE, the BRK instruction was executed and I was left in the monitor with the program counter set to \$360B.

After disassembling the code in the vicinity where the BRK occurred, I noticed at \$3620 the instructions

```
20 67 32 JSR $3967
00      BRK
```

This code struck me as a little strange since there should normally be executable code following a JSR instruction. Apparently, the subroutine at \$3967 would not be returning to its caller in the normal fashion.

### Some "Funny" Jumps

I disassembled the code from \$3967 and, at \$3980, I noticed an RTS instruction. Since the program would crash if it returned to the BRK just beyond JSR \$3967, I assumed that the author(s) of this code used the "funny jump" technique in an attempt to obscure their protection scheme. A "funny jump" occurs as a result of what happens when JSR and RTS instructions are executed. When a JSR is executed, values corresponding to the contents of the program counter + 2 are pushed on the stack (PHAed), high byte first and low byte second. When an RTS occurs, the two values are pulled (PLAed) off the top of the stack and transferred to the program counter, which is incremented by one before execution starts back up. If the stack pointer is disturbed after a JSR instruction, but before an RTS occurs, the program will continue its execution at the address determined by the values which are pulled off the stack when an RTS finally does occur. This "funny jump" technique is used quite a bit in the code which comprises DOS 3.3.

I next needed to locate the position at which the program would continue its execution when the RTS at \$3980 was executed. I made another copy of SS-ProDOS and changed this RTS to a BRK (after finding the code with the disk search utility) with my sector editor. When I booted up this copy of SS-ProDOS, it crashed into the monitor at \$3982 with a register display of

A=6B X=0B Y=00 P=31 S=F5

Although the stack pointer contained a

\$F5, I knew from experience that the routines which handle a BRK instruction will decrement the stack pointer by four. Therefore, the values from \$1FA-1FB would end up in the program counter upon the execution of the RTS. The values at \$1FA-1FB were 26 36 which meant that the execution would continue at \$3627.

### The Nibble Count Routine

When I listed the code from \$3627 it did not seem to be the nibble count routine for which I was searching, but there was an RTS instruction at \$3669. I again recopied SS-ProDOS and, after finding this RTS on the disk, I changed it to a BRK. After booting this disk, I was eventually left in the monitor at \$366A with the stack pointer holding a \$F7. This meant that the RTS would direct the program flow to the address corresponding to the values in \$1FC-1FD (which were 19 49). This indicated that the program would continue its execution from \$491A. When I disassembled from \$491A, I was pleased to find the following code:

```
491A- D0 13      BNE $492F
491C- 20 C1 22 JSR $22C1
491F- 29 3D      AND #3D
4921- 14         ???
4922- A3         ???
4923- 20 4F 22 JSR $224F
4926- 20 3D 22 JSR $223D
4929- 20 8E 22 JSR $228E
492C- 4C 12 49 JMP $4912
492F- AE 75 A5 LDX $A575
4932- 8D 89 C0 LDA $C089,X
4935- AD 43 A3 LDA $A343
4938- C9 01      CMP #01
.
.
.
.
4985- A9 FF      LDA #FF
4987- 8D 01 20 STA $2001
498A- A9 AE      LDA #AE
498C- 8D 27 04 STA $0427
498F- 4C 57 49 JMP $4957
49C2- A9 BA      LDA #BA
49C4- 8D 27 04 STA $0427
49C7- A9 00      LDA #00
49C9- 8D 01 20 STA $2001
49CC- CE 02 20 DEC $2002
49CF- D0 86      BNE $4957
49D1- BD 88 C0 LDA $C088,X
49D4- A9 A0      LDA #A0
49D6- 8D 27 04 STA $0427
49D9- 60         RTS
```

I say that I was pleased to find this code because instructions like

```
LDA $C08C,X
```

are utilized for direct reading of the disk and can usually be found in nibble count routines. To see if this nibble count routine could be disabled, I decided to place an

RTS at \$492F. To do this, I again needed the help of my disk search utility and sector editor.

This time when I booted the altered copy of SS-ProDOS, everything worked fine except that the message, "RAM MEMORY FAILURE F755", was displayed. However, after I hit the RETURN key, the Sensible Speller menu was displayed. From there on the program seemed to work just like original. I tested both the original and copied SS-ProDOS disks on several files that were generated by Appleworks and was very pleased that the exact same results were produced by both disks. That "RAM MEMORY FAILURE" message was still bothering me a bit, though.

### The Checksum Routine

Although I won't go into the details of the search, I eventually tracked down the routine which was generating this message and found that it resides at \$4A63. This routine performs a checksum of the contents of memory above \$2346. If the checksum is not what is supposed to be, the memory failure message is printed. Although changing the contents of just one memory location will cause the message to be printed, I found that the routine could be disabled by changing \$4A63 to a 60 (RTS). This checksum routine could be left enabled so that you would be alerted in the event that a memory failure actually occurred. In this event, you should take note of the value that is displayed with the "RAM MEMORY FAILURE" message. You will know that something really has gone wrong if an unusual checksum value is displayed. A partial listing of this checksum routine is reproduced below followed by the step-by-step instructions for making a backup of Sensible Speller for ProDOS.

```
4A63- A9 00      LDA #00
4A65- 8D 00 A3 STA $A300
4A68- 8D 01 A3 STA $A301
4A6B- AD 03 22 LDA $2203
4A6E- 85 00      STA $00
4A70- AD 04 22 LDA $2204
4A73- 85 01      STA $01
4A75- A0 00      LDY #00
4A77- 81 00      LDA ($00),Y
4A79- 4D 00 A3 EOR $A300
4A7C- 8D 00 A3 STA $A300
4A7F- C8         INY
4A80- 81 00      LDA ($00),Y
4A82- 4D 01 A3 EOR $A301
4A85- 8D 01 A3 STA $A301
4A88- E6 00      INC $00
4A8A- D0 02      BNE $4A8E
4A8C- E6 01      INC $01
4A8E- A5 01      LDA $01
4A90- CD 06 22 CMP $2206
```

Continued on next page



# ADVENTURE TIPS ADVENTURE TIPS

## \* Zork III

Infocom, Inc.

To win this game you must have patience, compassion, mercy, ingenuity, logic and perseverance.

Who sails the ocean? What do you say to him?

Look out if you get caught between a rock and a wet place.

A guiding principle is, "Thou shalt not kill."

Man does not live by bread alone, but one man may want bread alone.

\* Contributed By Cullen Johnson

## The House Of Mystery

Sierra On-Line

Before it gets too dark, light candle.

Need some water to put out the fire?

First you need the pitcher from the frig.

Then, turn faucet on.

Don't light the stove or you'll blow yourself up!

The picture looks suspicious. What can you use to pry it away from the wall?

## Softporn Adventure

Sierra On-line

Be sure to check out the reading material.

Examine all objects around you. They may hold clues.

Time for some TV? See what's on channel 6.

If you're gay, you won't be too happy with this game.

## Coveted Mirror

Penguin Software, Inc.

The safest way to get out of your cell is through a hidden passage.

To walk the corridors of the castle, you'll need to become invisible. Check out the round tower.

Boris wants you to give him things, but he also has something to offer.

Read the magician's book. You'll find a great deal of important information within its pages.

## Mask Of The Sun

Ultrasoft

The old woman is hungry. What do you have to offer?

Don't trade your valuable items to the peddler. You don't need what he has to offer.

Stuck in the platform room? Search around the platform and then open the door.

You don't have to be a ghost buster to get rid of the ghost in the sarcophagus. Just wait and he'll disappear on his own.

Skeletons are another story. Try hitting them with your amulet.

Search carefully in the room with the star. You're bound to "unlock" the mystery of your next move.

## \*\* Deadline

Infocom

Make sure you attend the reading of the will.

George can be rattled if you upset him early.

Count the tea cups.

Follow George if he's worried.

The garden is very interesting.

\*\* Contributed by Paul Below

## † Castle Wolfenstein

Muse Software

Holding the space bar down (and REPEAT key, if necessary) will shorten the time taken to open the chests.

If you want to avoid the guards but get at a chest, just go to an opening at the bottom of the screen (if there is one), point your gun into it and search.

Never point down to hold up a guard.

## † Sherwood Forest

Phoenix Software

Only a very "light" man can jump off a cliff and, for that matter, push a boulder.

The clothes make the man if you're Robin Hood of Sherwood Forest.

For those with a bad mouth, "Sorry" will do wonders.

† Contributed by Jeff Beard

## †† Blade Of Black Poole

Sirius Software

A drunk sea monster is a happy sea monster.

If you aren't wearing a helmet, you might bump your head.

Join the bird in singing.

The recluse has something to put on the altar.

†† Contributed by Michael Pocrass

cont...

Continued from previous page

4A93-	D0 05	BNE	\$4A9A
4A95-	A5 00	LDA	\$00
4A97-	CD 05 22	CMP	\$2205
4A9A-	90 09	BCC	\$4A75
4A9C-	A5 01	LDA	\$01
4A9E-	C9 3E	CMP	#\$3E
4AA0-	D0 04	BNE	\$4AA6
4AA2-	A5 00	LDA	\$00
4AA4-	C9 6C	CMP	#\$6C
4AA6-	90 E0	BCC	\$4A88
4AA8-	A5 01	LDA	\$01
4AAA-	C9 72	CMP	#\$72
4AAC-	D0 04	BNE	\$4AB2
4AAE-	A5 00	LDA	\$00
4AB0-	C9 B9	CMP	#\$B9
4AB2-	90 C1	BCC	\$4A75

etc.

## Making the Backup

1) Use COPYA or any other copy program to make a copy of Sensible Speller-ProDOS.

2) To disable the nibble count routine, search for a byte sequence of AE 75 A5 BD 89 C0 with your disk search utility. Search the entire disk for this sequence. It may be on more than one sector. On my copy of SS-ProDOS, I found this sequence on track \$A, sector \$E at byte \$2F. The AE should be changed to a 60 with a sector editor and written back to the disk.

3) To disable the routine that generates the "RAM MEMORY FAILURE" message, use your disk search utility to search for a sequence of A9 00 8D 00 A3 8D 01 A3. Search this entire disk, also. On my copy of SS-ProDOS, the code was on track \$A, sector \$D at byte \$63. Change the A9 in the sequence to a 60 with a sector editor and write it back to the disk.

4) Backups produced using this procedure can be copied by any copy program.

Many thanks to Bob and Hank for their assistance in the preparation of this article.



Want to order  
**BACK ISSUES?**

See page 29 for details.

**Requirements:**  
 Super IOB v1.2  
 48K Apple ][ Plus or equivalent

**S**uper IOB has proved itself to be a very useful program for normalizing copy-protected disks. However, user-friendliness is not one of its strong points. Writing a controller routine can be a real nuisance, often because of the need to perform several hex to decimal conversions. To alleviate these inconveniences, I decided to write an Applesoft program that would generate custom Super IOB controllers based upon several input parameters. I call this program The Controller Writer and have found that using it, in turn, makes Super IOB a lot more

convenient to use. The Controller Writer program (begins on page 17) requires that the user input several parameters based upon the protection scheme being used. The tracks and sectors to read and write are required inputs. Parameters such as changed address/data marks, the use of a foreign RWTS, and sector edits can also be specified. The controllers generated are stored to disk in the form of a compressed Text file.

I should mention that The Controller Writer does not automatically write Super IOB controllers for you; you still need some skills to identify the protection scheme being used. Such skills are best acquired through the use of a disk nibble utility such as The CIA's Linguist module or Diskview, lots of practice with protected disks, and the reading of Hardcore COMPUTIST. There are also certain forms of copy-protection for which The Controller Writer cannot generate controllers. The limitations of the program will be described later in this article.

### Using The Controller Writer

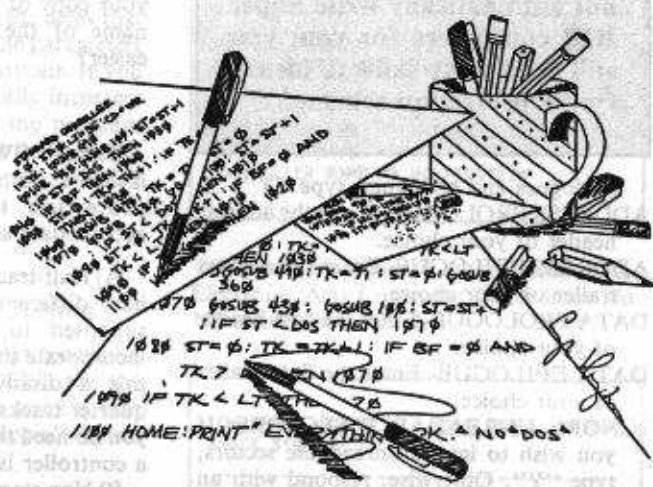
Before entering The Controller Writer (hereafter referred to as 'CW') listing, be sure to type 'NEW' or 'FP'. After you have entered the program and saved it on disk, type "RUN" to try it out. It will give you a title page and you must hit a key to start or the ESC key to end. Depending on the complexity of the desired controller, about 21 questions must be answered. All parameters that give you a "\$" as a prompt must be filled in with HEX values only. If no prompt is given (Track Step and Sector Step), then a decimal value or even a legal BASIC expression is allowed. All questions must be answered by typing "Y" for yes or "N" for no. When the hex input is two keystrokes, no RETURN is needed. One digit numbers must either be preceded with a zero or followed by RETURN. Backspacing is also allowed.

I have divided the input parameters into five sets: tracks to read and write, sectors to read, sectors to write, RWTS alterations, and controller identification. At the end of every set, the question "CORRECT?" will be asked. If "Y" is typed, CW will go on to the next set; otherwise, that set will be asked again. Pressing <sup>CTRL</sup>C at any time will rerun the program for you.

Here is a list of the parameters with a description of what each one means:

#### Set One: Tracks to Read & Write

- LOW TRACK-** This is the first or lowest track that will be read and written.
- HIGH TRACK-** The highest or last track to be read and written is entered here.
- TRACK STEP-** The legal BASIC expression you enter here will be added to the last track read in to get the next track. A value of 1 will read every track, a value of 2 will



# The Controller Writer

Super IOB is a proven time-saver when it comes to backing-up copy-protected disks. Using The Controller Writer, you'll find that the simplified generation of custom controllers for Super IOB will make the job of deprotecting programs even easier.

By Nick Galbreath



do every other track and an expression like 1+(TK=3) will read every track except track four.

### Set Two: Sectors to Read

**LOW SECTOR-** This will be the first sector read on every track. If you enter 02, then the controller will start by reading sector 2 on each track. This is normally 0.

**HIGH SECTOR-** This will be the last sector read on every track. For a 16 sector disk, enter an 0F; for 13 sectors, enter 0C. For Castle Wolfenstein, because it is 13 sectors and the sectors step by 2, you would enter a value of 18 (SC + SC = \$18).

C  
O  
R  
E

**SECTOR STEP-** The expression here will be added to the last sector read. Normally this should be 01, but for games like Castle Wolfenstein (which has the same protection as The Voice and Robot War), it will be 02. If you wanted to skip sector eight of every track, you would enter 1+(ST=7) and if you wanted to skip just sector 0C of track 9, you would enter 1+(ST=11 AND TK=9).

An example:

Low Sector: 3

High Sector: 0D

Sector Step: 2

These sectors and only these sectors will read on every track: 3,5,7,9,B,D

### Set Three: Sectors to Read

The parameters in this set are identical to those in set two except that these parameters refer to the writing out of the sectors. Your inputs will almost always be the normal DOS 3.3 values for this set of parameters since you generally will want to normalize copy-protected disks.

An example:

Low Sector: 00

High Sector: 0F

Sector Step: 01

"Super IOB has proved itself to be a very useful program for normalizing copy-protected disks. However, user friendliness is not one of its strong points...and I decided to write an Applesoft program that would generate custom Super IOB controllers..."

This will write out sectors 0 through 15 on every track. The number of sectors written on each track MUST be equal to the number of sectors read from every track. If they aren't, CW will assume that you made an error and will go back to set two.

### Set Four: RWTS Alterations

**USE FOREIGN RWTS-** This means, "Do you want a swap controller?" or, "Do you want to use a different RWTS routine?" Answer this question with a "Y" or an "N".

**ENTER NAME-** This will be the name of the new RWTS routine if you answered "Y" to the above.

**CHANGE SECTOR MARKS-** You will be asked this question only if a foreign RWTS routine is not going to be used. If you want to change the address/data

"...the Controller Writer does not automatically write Super IOB controllers for you; you still need some skills to identify the protection scheme..."

prologues and epilogues, type "Y".

**ADDRESS PROLOGUE-** Enter the address header of your choice.

**ADDRESS EPILOGUE-** Enter the address trailer of your choice.

**DATA PROLOGUE-** Enter the data header of your choice.

**DATA EPILOGUE-** Enter the data trailer of your choice.

**IGNORE UNREADABLE SECTORS-** If you wish to ignore unreadable sectors, type "Y". Otherwise, respond with an "N".

### Set Five: Controller Identification

**FIRST LINE REM-** Enter the text of the REM statement you want at line number 1000. The suffix "CONTROLLER" will be added to what you type in.

**VER. # OR DATE-** If you want the date or version number added to your line 1000 REM statement, enter it.

**CONTROLLER NAME-** This will be the name of the saved controller. A ".CON" will be suffixed to what you type in.

When you have completed your answers to the above questions, the words "SCREEN DUMP?" will appear at the bottom of the screen. When you answer yes to this question, CW will print the contents of the screen to a printer in slot 1. As the printing progresses, the screen will be turned to inverse characters. This will alleviate the boredom that might occur while the printing takes place. This inverting traumatizes some printers, so you may have to remove the INVERSE command from line 480. This entire routine lives at lines 410-500.

### Sector Edits

After the first SCREEN DUMP prompt you will be asked for the number of sector edits. You may include up to 9 sector edits.

If you don't want any performed, type a 0. Each sector edit is comprised of the Track, Sector and Byte number to be changed. The TO prompt asks for the new hexadecimal value of the particular byte within the specified sector. When you are finished typing in all the sector edits, you will once again have the opportunity to dump the screen to a printer.

### Hop To It

After answering all of those questions, insert the disk that you want the controller to be saved on, and press any key. The screen will be cleared and CW will now create the controller and save it to the disk. Presto! All done. To use the newly saved controller, load your copy of SUPER IOB and EXEC the name of the controller. What could be easier?

### Limitations

Though CW will write most controllers, it can't do everything. Some forms of protection that CW cannot directly accommodate are:

A) Half-tracks: Since half-tracks normally have different track numbers than they're supposed to, it would be difficult to incorporate them into CW. Don't consider this a disadvantage, though. Half and quarter tracks are used rarely and, even if you do need them, half the battle in writing a controller is done for you!

B) Non-standard sectors and tracks: Some protection schemes use non-standard sectors and tracks. Examples include Wayout, Type Attack and other Sirius disks which use a one-sector track. These can possibly be deprotected with the use of the RWTS routine that accompanies that disk.

C) More than one protection on a disk: Penguin Software is a good example of a company that does this. One protection technique is used on all even tracks and another is used for all odd tracks. See the Minit Man controller in Hardcore COMPUTIST No. 9 for instructions on how to do this type of controller. Another example is E-Z Draw 3.3 (also from Sirius) which has 8 different address headers per track (1 header for every 2 sectors on each track).

### Notes On Program Design

CW has some interesting subroutines that you might want to use in your own programs. The screen dump routine, which lives from line 410 through line 500, is written entirely in BASIC. It uses the low resolution command "SCRN" to determine which characters are on the screen. Line 480 contains an INVERSE statement. As I mentioned, this might mess up your printer. It works fine on my Apple Scribe.

The error handling routine checks to see if an Applesoft or DOS error has occurred. If an Applesoft error has occurred, it prints

the line of the error and the error that occurred there. If the error was generated by DOS, it goes to the correct routine to try again. This error handling routine lives at lines 1510 to 1540.

At lines 130-220 lives the 1 hex byte input routine. This routine uses the string called "HXS" for the conversion. It scans the string for what you have typed in. If it doesn't exist in "HXS", it ignores what was typed; otherwise, it adds the decimal number to variable "A."

The evaluate expression routine which lives at lines 520-600 is probably the most bizarre of all the routines in the program. It creates an artificial line number starting at \$200 which will evaluate the number of sectors on a track using the sector step expression. This routine causes the delay that you may notice after entering the parameters for the sectors to be read or written. If you type in an illegal or syntactically incorrect expression for a sector step, the program may hang or bomb. To recover the program (which will LIST as garbage), press RESET and then type:

POKE 104,8:RUN

### Controller Writer In Action

For an example of the type of controller which can be generated by CW, see the screen dump below from the session that created a controller for Sublogic's Space Vikings program:

```

LOW TRACK => $03    HIGH TRACK => $22
TRACK STEP => 1

-----
READING:
LOW SECTOR => $00    HIGH SECTOR => $0F
SECTOR STEP=> 1

-----
WRITING:
LOW SECTOR => $00    HIGH SECTOR => $0F
SECTOR STEP=> 1

-----
ADDRESS PROLOGUE (D5AA96) => $D5AA96
ADDRESS EPILOGUE (DEAA) => $DEAA
DATA PROLOGUE (D5Aaad) => $D5AADA
DATA EPILOGUE (DEAA) => $EDAA
IGNORE UNREADABLE SECTORS? => NO

-----
FIRST LINE REM => SPACE VIKINGS
VER. # OR DATE => 1.5
CONTROLLER NAME => VIKINGS

```

Although it was not a run-away best seller, Space Vikings demonstrates some of CW's features nicely. One can tell at a glance which method of protection is being used. A 'Low track' of 3 indicates that a modified DOS is our most likely candidate. The data prologue has been changed to D5AADA from the normal D5AAAD, and the data epilogue has been changed to EDAA from the usual DEAA. Plain and simple. To make this

controller, just type the above screen dump into CW - no programming today!

The first screen dump during the creation of a controller for Castle Wolfenstein would look like this:

```

LOW TRACK => $03    HIGH TRACK => $22
TRACK STEP => 1

-----
READING:
LOW SECTOR => $00    HIGH SECTOR => $18
SECTOR STEP=> 2

-----
WRITING:
LOW SECTOR => $00    HIGH SECTOR => $0C
SECTOR STEP=> 1

-----
FOREIGN RWTS => RWTS.13
IGNORE UNREADABLE SECTORS? => YES

-----
FIRST LINE REM => CASTLE WOLFENSTEIN
VER. # OR DATE =>
CONTROLLER NAME => WOLFENSTEIN

```

The screen dump to create most of the Electronic Arts Controller (from Issue No. 13) would look like this:

```

LOW TRACK => $0    HIGH TRACK => $22
TRACK STEP => 1

-----
READING:
LOW SECTOR => $00    HIGH SECTOR => $0F
SECTOR STEP=> 1

-----
WRITING:
LOW SECTOR => $00    HIGH SECTOR => $0F
SECTOR STEP=> 1

-----
ADDRESS PROLOGUE (D5AA96) => $D5AA96
ADDRESS EPILOG (DEAA) => $DEAA
DATA PROLOGUE (D5Aaad) => $D5B8CF
DATA EPILOGUE (DEAA) => $DEAA
IGNORE UNREADABLE SECTORS? => NO

-----
FIRST LINE REM => ELECTRONIC ARTS
VER. # OR DATE => 1.0
CONTROLLER NAME => EOA

NUMBER OF SECTOR EDITS? => 5

TRACK- $01 SCTR- $0F BYTE- $68 TO- $18
TRACK- $01 SCTR- $0F BYTE- $69 TO- $60
TRACK- $01 SCTR- $0F BYTE- $6A TO- $EB
TRACK- $03 SCTR- $47 BYTE- $47 TO- $AA
TRACK- $03 SCTR- $51 BYTE- $47 TO- $AD

```

The resulting controller would then require that an "IF TK > 3 THEN" be inserted in line 1020 just before the GOSUB 190. It would then work exactly like the controller in Hardcore COMPUTIST No. 13.

I hope that those of you who enter and use

The Controller Writer will find it a worthy addition to your library of disk utilities. Comments on the program and any enhancements you would like to see added to it are welcome.

### Controller Writer

```

10 REM /-----/!
20 REM / CONTROLLER / !
30 REM / WRITER / !
40 REM ----- !
50 REM ! BY: NICK ! !
60 REM ! GALBREATH ! /
70 REM ! HC NO. 16 ! /
80 REM -----/
90 REM
100 TEXT : NORMAL : SPEED= 255 : CLEAR
: ONERR GOTO 1510
110 GOTO 650
120 REM 1 BYTE INPUT
130 PRINT "$";
140 GET AS : FOR J = 1 TO 17 : IF AS <>
MID$(HX$,J,1) THEN NEXT : GOTO
140
150 IF J = 17 THEN RUN
160 PRINT AS; : A = (J - 1) * 16
170 GET AS : IF AS = CH$ THEN PRINT AS;
: GOTO 140
180 IF AS = CM$ THEN A = A / 16 : PRINT
"@"; : RETURN
190 FOR J = 1 TO 17 : IF AS <> MID$(HX$,
J,1) THEN NEXT : GOTO 170
200 IF J = 17 THEN RUN
210 PRINT AS; : A = A + J - 1
220 RETURN
230 REM GET A STRING
240 BS$ = "" : A = PEEK (36) + 1
250 GET AS : IF AS = CC$ THEN RUN
260 IF LEN (BS) = LN AND AS <> CM$ AND
AS <> CH$ THEN 250
270 IF AS = CM$ AND BS <> NL$ THEN AS =
BS : RETURN
280 IF AS = CH$ AND BS = "" THEN 250
290 IF AS = CH$ THEN PRINT AS " " AS; :
IF LEN (BS) = 1 THEN BS = "" : GOTO
250
300 IF AS = CH$ THEN BS = LEFT$ (BS ,
LEN (BS) - 1) : GOTO 250
310 IF AS < SP$ THEN 250
320 PRINT AS; : BS = BS + AS : GOTO 250
330 REM GET A "Y" OR "N"
340 GET AS : IF AS <> "Y" AND AS <> "N"
AND AS <> CC$ THEN 340
350 IF AS = CC$ THEN RUN
360 IF AS = "Y" THEN PRINT "YES"; :
RETURN
370 PRINT "NO"; : RETURN
380 REM PRINT BANNER OF "--" 'S
390 FOR I = 1 TO 40 - PEEK (32) : HTAB I
: PRINT "--" : NEXT : RETURN
400 REM SCREEN DUMP ROUTINE
410 VTAB 24 : HTAB 1 : CALL HM : PRINT
"SCREEN^DUMP?" : GOSUB 340 :
IF AS = "N" THEN RETURN
420 VTAB 24 : HTAB 1 : CALL HM : VTAB
23 : PRINT CD$ "PR#1"
430 FOR B = 0 TO 40 STEP 2 : REM ENTER
LINES TO DUMP TIMES TWO

```



```

440 FOR A=0 TO 39
450 HTAB A+1:VTAB INT(B/2)+1
460 C=SCRN(A,B)+SCRN(A,B+1)*
16
470 IF C<160 THEN C=C+64:GOTO 470
480 AS=CHR$(C):INVERSE:PRINT
AS:;NORMAL
490 NEXT:PRINT CMS:;NEXT
500 NORMAL:PRINT CDS"PR#0":
RETURN
510 REM EVALUATE EXPRESSION
520 AS="A=A+1:ST=ST+"+BS+
":IFST<" +STR$(HS) + "THEN5"
:BS="":ONERR GOTO 540
530 FOR A=1 TO LEN(AS):POKE 511+A
,ASC(MID$(AS,A,1)):NEXT:
POKE 511+A,0:CALL 768
540 FOR A=510 TO 768:IF PEEK(A)
THEN BS=BS+CHR$(PEEK(A)):
NEXT
550 POKE 512,0:BS=517+LEN(BS):
POKE 512,0:POKE 513,B-512:
POKE 216,0
560 POKE 514,2:POKE 515,5:POKE
516,0:ONERR GOTO 510
570 FOR A=1 TO LEN(BS):POKE 516+A
,ASC(MID$(BS,A,1)):NEXT
580 A=516+A:POKE A,0:POKE A+1
,1:POKE A+2,8:POKE A+3,7
590 POKE A+4,0:POKE A+5,177:
POKE A+6,0:POKE 104,2
600 A=0:ST=LS:GOSUB 5:POKE 104
,8:RETURN
610 REM CHECK IF SET IS OK
620 PRINT:CALL HM:PRINT
"CORRECT?":;GOSUB 340:IF AS
="Y" GOTO 390
630 RETURN
640 REM INITIALIZE VARIABLES
650 CC$=CHR$(3):CDS=CHR$(13)+
CHR$(4):CG$=CHR$(7)
660 QS=CHR$(34):HX$=
"0123456789ABCDEF"+CC$
670 CMS=CHR$(13):CH$=CHR$(8):
FOR A=768 TO 776:READ LT:POKE
A,LT:NEXT
680 PRINT CDS"MONONC" CDS"MONIO":
HOME:SP$="":LN=25:HM=-
958
690 REM TITLE PAGE
700 VTAB 10:PRINT SPC(12)
"CONTROLLER^WRITER":VTAB 12
710 PRINT SPC(19)"BY":VTAB 14:
PRINT SPC(13)"NICK^
GALBREATH"
720 VTAB 17:PRINT SPC(3)
"COPYRIGHT^1984,^HARDCORE^
COMPUTIST":VTAB 22
730 PRINT SPC(9)"PRESS^ANY^KEY^
TO^BEGIN":VTAB 23:PRINT SPC(
14)"<ESC>^TO^QUIT"
740 WAIT-16384,128:GET AS:IF AS
=CHR$(27) THEN END
750 REM SET ONE
760 HOME:GOSUB 390:PRINT "LOW^
TRACK^>":;GOSUB 130:LT=A
770 HTAB 22:PRINT "HIGH^
TRACK^>":;GOSUB 130:HT=A
780 PRINT:PRINT "TRACK^STEP^>";

```

```

:GOSUB 240:TSS=AS
790 GOSUB 620:IF AS="N" THEN 760
800 REM SET TWO
810 VTAB 5:CALL HM:PRINT TAB(17)
"READING:"
820 PRINT "LOW^SECTOR^>":;GOSUB
130:L1=A
830 HTAB 22:PRINT "HIGH^
SECTOR^>":;GOSUB 130:H1=A
+1
840 PRINT:PRINT "SECTOR^STEP^>":
:GOSUB 240:GOSUB 620:IF AS=
"N" THEN 810
850 S1$=BS:HS=H1:LS=L1:GOSUB
520:NS=A
860 REM SET THREE
870 VTAB 9:CALL HM:PRINT TAB(17)
"WRITING:" :PRINT "LOW^
SECTOR^>":;GOSUB 130
880 L2=A:HTAB 22:PRINT "HIGH^
SECTOR^>":;GOSUB 130:H2=A
+1
890 PRINT:PRINT "SECTOR^STEP^>":
:GOSUB 240:GOSUB 620:IF AS=
"N" THEN 870
900 S2$=BS:HS=H2:LS=L2:GOSUB
520:IF A<>NS THEN 810
910 REM SET FOUR
920 VTAB 13:HTAB 1:CALL HM:PRINT
"USE^FOREIGN^RWTS?^>":;
GOSUB 340:IF AS="N" THEN 950
930 POKE 34,15:LD=1:PRINT:INPUT
"ENTER^NAME^":R1$
940 TEXT:VTAB 13:CALL HM:PRINT
"FOREIGN^RWTS^>":R1$:GOTO
1020
950 SM=1:VTAB 13:HTAB 1:CALL HM:
PRINT "CHANGE^SECTOR^
MARKS?^>":;
960 GOSUB 340:PRINT:IF AS="N"
THEN SM=0:GOTO 1020
970 VTAB 13:CALL HM:PRINT
"ADDRESS^PROLOGUE^
(D5AA96)^>":;
980 GOSUB 130:A1=A:GOSUB 140:A2=
A:GOSUB 140:A3=A
990 PRINT:PRINT "ADDRESS^
EPILOGUE^(DEAA)^>":;
GOSUB 130:A4=A:GOSUB 140:A5=
A
1000 PRINT:PRINT "DATA^PROLOGUE^
(D5AAAD)^>":;GOSUB 130
:D1=A:GOSUB 140:D2=A:GOSUB
140:D3=A
1010 PRINT:PRINT "DATA^EPILOGUE^
(DEAA)^>":;GOSUB 130
:D4=A:GOSUB 140:D5=A:PRINT
1020 PRINT "IGNORE^UNREADABLE^
SECTORS?^>":;GOSUB 340:IF
AS="Y" THEN US=1
1030 GOSUB 620:IF AS="N" THEN 920
1040 REM SET FIVE
1050 V=PEEK(37)+1:SP$="":LN=
20:NLS=CC$
1060 VTAB V:HTAB 1:CALL HM:PRINT
"FIRST^LINE^REM^>":;GOSUB
240:R2$=AS+"^CONTROLLER"
1070 PRINT:PRINT "VER.^#^OR^
DATE^>":;GOSUB 240:IF AS<
>" THEN R2$=R2$+"^>"+AS

```

```

1080 PRINT:CALL HM:PRINT
"CONTROLLER^NAME^>":;GOSUB
240
1090 IF LEFT$(AS,1)<"A" THEN
PRINT CG$:VTAB V+1:GOTO 1080
1100 CNS=AS+"^CON":VTAB V+2:
GOSUB 620:IF AS="N" THEN 1060
1110 REM SCREEN DUMP
,SECTOR EDITS
1120 GOSUB 410:HOME:PRINT
"NUMBER^OF^SECTOR^EDITS^>":
:LN=1:SP$="":GOSUB 240
1130 PRINT:PRINT:SE=VAL(AS):IF
SE=0 THEN 1200
1140 DIM TK(SE),ST(SE),BY(SE)
,OT(SE):FOR B=1 TO SE
1150 V=PEEK(37)+1
1160 VTAB V:CALL HM:PRINT
"TRACK^":;GOSUB 130:TK(B)=
A:PRINT "SECTOR^":;GOSUB
130:ST(B)=A
1170 PRINT "BYTE^":;GOSUB 130
:BY(B)=A:PRINT "ATO^":;
GOSUB 130:OT(B)=A
1180 GOSUB 620:HTAB 1:IF AS="N"
THEN 1160
1190 VTAB V+2:NEXT B:GOSUB 410
1200 VTAB 24:HTAB 1:CALL HM:PRINT
"INSERT^DISK^AND^HIT^A^KEY^":
;
1210 GET AS:IF AS=CC$ THEN RUN
1220 HOME:PRINT "STAND^
BY...WRITING^CONTROLLER...":
VTAB 3
1230 REM WRITE THE CONTROLLER
1240 PRINT CDS"OPEN" CNS:CDS
"DELETE" CNS:CDS"OPEN" CNS:CDS
"WRITE" CNS
1250 PRINT "DEL1000,9999"
1260 PRINT "1000REM" R2$
1270 PRINT "1010TK=" LT":LT=" HT+1
":CD=WR:MB=";
1280 PRINT NS*7+39:;IF US THEN
PRINT":ONERRGOTO550";
1290 PRINT:PRINT "1020ST=" L1
":T1=TK:GOSUB490";
1300 IF SM THEN PRINT
":RESTORE:GOSUB190:GOSUB210:
GOSUB170";
1310 IF LD THEN PRINT":GOSUB360";
1320 PRINT:PRINT "1030GOSUB430:
GOSUB100:ST=ST+" S1$:IFST<"
H1" THEN1030"
1330 PRINT "1040IFBFTHEN1060"
1340 PRINT "1050ST=" L1":TK=TK+"
TSS":IFTK<LTTHEN1030"
1350 PRINT "1060":;IF SE THEN PRINT
"GOSUB310:";
1360 IF LD THEN PRINT "GOSUB360:";
1370 IF SM THEN PRINT "GOSUB230:";
1380 PRINT "TK=T1:ST=" L2
":GOSUB490"
1390 PRINT "1070GOSUB430:
GOSUB100:ST=ST+" S2$:IFST<"
H2" THEN1070"
1400 PRINT "1080ST=" L2":TK=TK+"
TSS":IFBF=0ANDTK<LTTHEN1070"
1410 PRINT "1090IFTK<LTTHEN1020"
1420 PRINT "1100HOME:AS=" QS"ALL^
DONE" QS":GOSUB450:END"

```

```

1430 IF SM THEN PRINT "5000DATA" A1
      "A2","A3","D1","D2","D3
      "A4","A5","D4","D5
1440 IF SE=0 THEN 1470
1450 PRINT "5010DATA" SE "A CHANGES"
      : FOR A=1 TO SE
1460 PRINT 5010+A*10 "DATA" TK(A)
      "ST(A)","BY(A)","OT(A) :
      NEXT
1470 IF LD THEN PRINT
      "10010?CHRS(4)" QS "BLOAD" R1$
      "A$1900" QS
1480 PRINT "?" QS LEFT$(CNS,LEN
      (CNS)-4) "A CONTROLLER"
      ENTERED" QS
1490 PRINT CD$ "CLOSE" CNS : RUN
1500 REM ERROR TRAP
1510 NORMAL : A=PEEK(222)
1520 IF A=0 OR (A<255 AND A>15)
      THEN TEXT : PRINT : PRINT CG$
      "ERROR#" A "A IN LINE" PEEK
      (218)+PEEK(219)*256 : END
1530 IF A=255 THEN RUN
1540 VTAB 23 : HTAB 1 : PRINT CG$
      "WEIRD DOS ERROR#" A : GOTO
      200
1550 DATA 162,255,160,2,132,19
      ,76,108,213

```

### Checksums

10	- \$BADD	430	- \$D045
20	- \$9813	440	- \$75EF
30	- \$4D3B	450	- \$3822
40	- \$AD92	460	- \$12B0
50	- \$C899	470	- \$C193
60	- \$FF65	480	- \$1876
70	- \$A38F	490	- \$9D89
80	- \$A900	500	- \$CD52
90	- \$924D	510	- \$3A06
100	- \$90FC	520	- \$7851
110	- \$AE95	530	- \$742E
120	- \$E329	540	- \$51F1
130	- \$58E4	550	- \$A3CE
140	- \$523D	560	- \$FD3F
150	- \$E0DD	570	- \$6D0C
160	- \$7F29	580	- \$7B2F
170	- \$EBCB	590	- \$86BB
180	- \$958A	600	- \$5253
190	- \$C983	610	- \$B092
200	- \$D629	620	- \$D092
210	- \$5A4A	630	- \$5EE0
220	- \$7328	640	- \$277D
230	- \$3A89	650	- \$4590
240	- \$265A	660	- \$64C2
250	- \$0CC4	670	- \$D55C
260	- \$8D83	680	- \$47A4
270	- \$F14A	690	- \$73F5
280	- \$E3BD	700	- \$8E3A
290	- \$EE2C	710	- \$8EE2
300	- \$61FE	720	- \$8D73
310	- \$8E78	730	- \$9226
320	- \$505A	740	- \$3B30
330	- \$2ACD	750	- \$27C7
340	- \$08A6	760	- \$80A3
350	- \$2D68	770	- \$1F15
360	- \$A158	780	- \$1072
370	- \$0845	790	- \$66F2
380	- \$51BC	800	- \$3F90
390	- \$68DB	810	- \$6F08
400	- \$07CE	820	- \$2591
410	- \$C18F	830	- \$83A6
420	- \$0E79	840	- \$44CF

850	- \$D961	1040	- \$2872
860	- \$E833	1050	- \$D601
870	- \$DB48	1060	- \$2788
880	- \$26AA	1070	- \$C5BB
890	- \$67D5	1080	- \$1283
900	- \$A7A4	1090	- \$F79C
910	- \$81F2	1100	- \$A517
920	- \$EF16	1110	- \$8345
930	- \$307B	1120	- \$191A
940	- \$9A62	1130	- \$8A19
950	- \$01AE	1140	- \$24FE
960	- \$8386	1150	- \$8643
970	- \$295F	1160	- \$4411
980	- \$F2C0	1170	- \$F242
990	- \$A723	1180	- \$23D7
1000	- \$7C09	1190	- \$758E
1010	- \$E8FA	1200	- \$1206
1020	- \$FCFC	1210	- \$526C
1030	- \$3BA8	1220	- \$8624

1230	- \$8400	1420	- \$FE87
1240	- \$C34C	1430	- \$12FC
1250	- \$C986	1440	- \$8B48
1260	- \$339F	1450	- \$D2EB
1270	- \$8467	1460	- \$AD69
1280	- \$5562	1470	- \$C4A9
1290	- \$9A39	1480	- \$FB68
1300	- \$D6D0	1490	- \$3C37
1310	- \$4BB1	1500	- \$C1A8
1320	- \$3E7B	1510	- \$C9FC
1330	- \$8BEA	1520	- \$EFD8
1340	- \$3D2E	1530	- \$EDB4
1350	- \$1F83	1540	- \$7BFA
1360	- \$8734	1550	- \$544E
1370	- \$C973		
1380	- \$8B5E		
1390	- \$2E5B		
1400	- \$3FF2		
1410	- \$A28C		

## A Fix For The Beyond Castle Wolfenstein Softkey

Several readers have reported problems in performing the softkey for Beyond Castle Wolfenstein (Hardcore COMPUTIST No. 13), even after we belatedly printed the accompanying Super IOB controller for it in Hardcore COMPUTIST No. 14. If the original procedure and controller don't do the trick for you, try this revised procedure and controller instead.

1) Begin by booting Beyond Castle Wolfenstein.

### PR#6

2) The instant you hear your disk drive head move (other than the initial recalibration), press RESET.

3) Enter the monitor and move the Wolfenstein RWTS out of the way so we can boot a slave disk.

### CALL-151

```
1900 < B600.BFFFM
```

4) Boot your 48k slave disk (or Super IOB disk).

### 6<sup>CH</sup>P

5) Save the RWTS from Beyond Castle Wolfenstein.

```
BSAVE RWTS.BEYOND WOLF.
A$1900,LSA00
```

6) Remove the disk in the drive, put a blank disk in the drive and initialize it.

### INIT HELLO

7) Install the controller at the end of this correction into Super IOB and run it.

8) Follow the prompts, but *DO NOT* format the backup.

9) When the copy is done, clear the

program in memory

```
FP
```

10) Type this short greeting program.

```
10 POKE 1010,105 : POKE 1011,255 :
POKE 1012,90
20 PRINT CHR$(4) "BRUN@INIT"
```

11) Save it on the copy of Beyond Castle Wolfenstein.

### SAVE HELLO

### Beyond Castle Wolfenstein Controller

```
60 LOMEM:8960 : HIMEM:9983 : GOTO
10010
1000 REM BEYOND CASTLE
WOLFENSTEIN
1010 TK=3 : ST=0 : LT=35 : CD=WR
1020 T1=TK : GOSUB 490 : GOSUB
1110
1030 GOSUB 430 : GOSUB 100 : ST=ST
+1 : IF ST < DOS THEN 1030
1040 IF BF THEN 1060
1050 ST=0 : TK=TK+1 : IF TK < LT
THEN 1030
1060 GOSUB 490 : TK=T1 : ST=0 :
GOSUB 1110
1070 GOSUB 430 : GOSUB 100 : ST=ST
+1 : IF ST < DOS THEN 1070
1080 ST=0 : TK=TK+1 : IF BF=0
AND TK < LT THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "EVERYTHING"
O.K. AND DOS ON COPY" : END
1110 POKE 253,25 : POKE 255,182 :
POKE 224,10 : CALL 832 :
RETURN
10010 PRINT CHR$(4) "BLOAD"
RWTS.BEYOND WOLF,A$1900"
```



# The Lone Catalog Arranger

By Ray Darrah

## Requirements:

- Apple ][ Plus with slot 0 RAM card or Apple //e
- One disk drive with DOS 3.3
- An accurate typing hand

The following is the first part of a program called The Lone Catalog Arranger (henceforth referred to as LCA). LCA was designed for the manipulation of DOS 3.3 disk directories. With this program you can:

CORE

- 1) View the free and used space on a disk
- 2) Undelete, Delete, Lock and Unlock files
- 3) Insert and see illegal characters in filenames
- 4) Change the order of the files in the CATALOG
- 5) Remove deleted files from the directory
- 6) Create dazzling titles for your disks

The program is comprised of two parts: An Applesoft BASIC program, and a Machine Language program. *Due to space limitations, only the BASIC portion of the program is presented here. This program will not work without the machine language program that will appear in Hardcore COMPUTIST No. 17.* Please note that this program *does* require either an Apple //e or 48K Apple ][ Plus (or compatible) with a slot 0 RAM card.

Even though this program will not work without the companion machine language portion, you can become familiar with the following "how-to's" of the program.

## Typing It In

The first step in using a program is getting it into your computer. Because of the length of this program, it is very susceptible to typographical errors. BE CAREFUL. A typographical error could make the program trash a disk. I would strongly recommend using the Checksoft program (From Hardcore COMPUTIST No. 1 or The Best of Hardcore Computing) after typing it in. You should save the program with:

SAVE BAS.LONE ARRANGER

## Reading The Disk

When the program is RUN, the first thing you will see is the help screen (which doubles as a title page). At the bottom left of the screen, you will see the highlighted message, "INSERT DISK". You should then insert the disk with the directory you wish to modify. At the bottom center of the screen are the slot, drive and volume parameters. The default is slot 6, drive 1, volume 0 (volume 0 will match any volume). You should type the correct numbers for these prompts or you can press RETURN if the default is correct. After you press return for the volume number, the drive will start up and the directory will be read in.

## Editing The Directory

After a couple of seconds, the directory will be displayed on the screen. Files with a hyphen ("-") at the left edge of the screen have been deleted. The middle entry on the screen will be highlighted. This highlighted file is referred to by LCA as the "current file". It is the default file that a function will be performed on. To change the current file, either press the left arrow or right arrow. This will scroll the directory in the appropriate direction and a different file will be highlighted. For most of LCA's options, pressing the "C" key will perform that option upon the highlighted, or current file.

Another way to inform LCA that you want it to perform some action on a file is to type an "F", for filename. This will allow you to directly type in a filename, optionally with the use of the wildcard character ("=").

The use of the wildcard character will allow you to select files within the directory that have a particular pattern of characters in them. For example, if you type in "CON.=", LCA will search the directory for any files which begin with "CON." such as "CON.EOA", "CON.WOLFENSTEIN" or "CON.SAT". When a filename with the wildcard pattern is found, LCA will ask you if you want to perform the current function upon that file. Only a single wildcard character is allowed when entering a filename so, if you type the "=" as the first character of the filename, all files within the directory will be selected.

Below the directory is a command index line and the "command" prompt. You will notice that the left and right arrows are not in the command index line. Instead, the less than sign (<) and greater than sign (>) are displayed. This is because you may use the less than and greater than sign keys (or "<" and ">") as well as the left and right arrow keys. The arrow keys will, however, move you slightly faster than the less than and greater than keys.

## Allocation Map

By pressing "A" while the command prompt is displayed, you may view the free

and used sectors on the disk. This changes as files are deleted or undeleted. You may abort the function prematurely by pressing ESCape.

## Simulated Catalog

When "C" is pressed at the command prompt, the directory is displayed as if it were stored on the disk at that point. This is called a "simulated catalog" because it is read from the computer's memory and not directly from the disk. This "simulated catalog" will not be written to the disk until you inform LCA that it is OK to do so.

## Delete Files

At the command prompt, the typing of "D" will invoke the delete files routine. A list of the functions that you can perform on more than one file are similar in operation. The first thing you will be asked is whether you want to perform the function on the current file or on a specified file. If you press RETURN or "C" the function (in this case, deletion) will be performed on the current file. If you press "F", then you will be asked for a filename. You may use the "=" character in the filename as a wildcard.

When the file you specified is found, you will be asked if you want to perform the function (in this case, deletion) on it. The default (pressing RETURN) is yes. You may type "N" to skip the file or ESCape to abort the function. Do not be alarmed when the disk drive starts up, LCA only writes to the disk during the save directory function.

## Exhume Files

Pressing "E" while the command prompt is displayed will allow the undeletion of deleted file(s). This command follows the same structure outlined in the instructions for the delete option. Only files that have been deleted can be exhumed. When the disk drive starts up, again, don't be alarmed. Nothing is being written to the disk. Remember that a deleted file can only be properly exhumed if none of its sectors have been written over.

## Lock/Unlock Files

When "L" or "U" is pressed at the command prompt, the lock or unlock files routine is invoked. The procedure for locking and unlocking files is the same as the procedure outlined in the delete instructions. Only files that are locked can be unlocked and only files that are unlocked can be locked. You may also lock and unlock deleted files if you so desire.

## Move File

At the command prompt, the typing of "M" will invoke the move file routine. This use of this option is similar to that of the delete option in that you may only move the

current file or manually type in the name of the file you wish to move.

The wildcard character ("=") can also be used with the move option. If more than one wildcard match is found, you will be asked, "MOVE?". You must answer this question with a "Y" or "N". As soon as you answer to the affirmative, LCA will use this file as the object of the move.

Once the file to move is found, it will become highlighted. Pressing the left and right arrow keys will scroll the remainder of the directory around the highlighted file. When the highlighted file is in the position you want it in the directory, press RETURN or ESCape to place it there.

### Purge Deleted Files

Pressing "P" while the command prompt is displayed will permanently remove from the directory the current or specified file. Be careful when using this option, as once a file has been purged from the directory it can no longer be exhumed by the undelete option.

### Rename Files

When "R" is pressed at the command prompt, you will be allowed to rename the current or specified file(s). Normal, Inverse, Flashing, Control and Lowercase characters can be entered for either the original name or for the new filename. To place these characters in the filename, you must first hold down the CTRL key and the first letter of the type of characters you wish. For example, if you want to place a CTRL H in a filename, you would type CTRL C for control character and then H. Every character you type after the CTRL C will be converted into a control character. Likewise, every character you type after a CTRL F will be converted into a flashing character. To return to regular characters, type CTRL N for Normal. Needless to say, if you want to enter any lower case characters, your computer must have lower case capability. Pressing RETURN will chop off the filename at the cursor position. If you press RETURN while the cursor is on top of the first character, the filename will be left unchanged.

### Title File

At the command prompt, the typing of "T" will invoke the title file routine. This routine allows you to insert "dummy" files into the directory that can act as separators or as a dazzling title for your disk. The files it makes occupy no room on your disk (other than the space in one of the directory sectors). However, title files may be copied incorrectly by a file transfer program (such as FID) because of the manner in which LCA creates these title files.

First of all, the filename of the title file may be typed in using the technique described in the rename instructions. The file is then inserted into the directory at the

current file position and the move file routine is automatically invoked. This allows you to move the newly created file into the directory and then to any location in that directory.

### Saving The Directory

By pressing "ESC" at the command prompt, you will get a mini-menu. On this secondary menu, you have these options:

- 1) Save the directory in memory
- 2) Return for more editing
- 3) Restart the program, losing any changes

The default is to save the directory. The other options may be selected by pressing either "R" or "ESC".

### Limitations

For the most part, this program has no limitations. It will work fine if you have a more-or-less normal DOS 3.3 in the machine. However, in order to keep the machine language portion of the program as simple as possible, LCA cannot manipulate more than 85 files.

The assembly language portion of The Lone Catalog Arranger will appear in the CORE Section of next month's issue. For those who can't bear the wait, both the Applesoft and assembly language portions of The Lone Catalog Arranger are available on the Library disk for this issue of Hardcore COMPUTIST. Although not currently listed in the advertisement on page 27, the library disk for this issue will be available at the same price of \$9.95.

### The Lone Catalog Arranger Basic Program

```

10 REM
20 REM
30 REM \---/ \---/
40 REM \ LONE /
50 REM \ ARRANGER /
60 REM \---/ \---/
70 REM
80 REM
90 REM
100 PRINT CHR$(4) "MAXFILES1" :
    HIMEM: 32767 : CLEAR : GOTO 1300
110 REM MOVE FORWARD OR BACKWARD
120 IF CF > 0 THEN CF = CF - W
130 RETURN
140 IF CF < MX THEN CF = CF + W
150 RETURN
160 REM SCAN FOR A FILE
170 CF = 0 : FOR A = W TO 30 : IF MIDS
    (B$, A, W) <> WILD$ THEN NEXT
180 HOME : B$ = LEFT$(B$, A - (A > W))
190 HTAB W : X =USR (CF) : IF MIDS
    (F$(CF), 9, LEN (B$)) = B$ OR B$
    = WILD$ THEN X = W : RETURN
200 CF = CF + W : IF CF < MX + W THEN 190
210 CF = INT (MX / 2) : HOME : X =USR
    (CF) : X = 0 : RETURN
220 REM GET A FILENAME
230 A$ = SP$

```

```

240 C$ = A$ : FOR A = 0 TO 29 : POKE 540
    + A, ASC (MID$(A$, A + W, W)) :
    NEXT

```

```

250 VTAB 21 : HTAB 20 : CALL 64578 :
    PRINT "FILENAME" : PRINT : PRINT
    "NRM">A :

```

```

260 A = 128 : X = 128 : POKE 56157, 0 :
    POKE 54, 77 : POKE 55, 8 : CALL
    1002 : C = 0 : PRINT A$ :

```

```

270 VTAB 24 : HTAB 6 + C : PRINT CHR$(
    222) : : HTAB PEEK (36) : GET A$
    : PRINT CHR$(160) : : HTAB PEEK
    (36) : VTAB 23

```

```

280 IF (A$ <"^" OR A$ > CHR$(95))
    AND A$ <> C$ AND A$ <> CH$ AND A$
    <> CL$ AND A$ <> C1$ AND A$ <>
    CF$ AND A$ <> CN$ AND A$ <> CC$
    AND A$ <> ESC$ AND A$ <> CUS$ THEN
    270

```

```

290 IF A$ = ESC$ THEN PR# 0 : CALL 1002
    : POKE 56157, 128 : RETURN

```

```

300 IF A$ = C1$ THEN A = -64 : X = 0 : A$
    = "INV" : GOTO 430

```

```

310 IF A$ = CN$ THEN A = 128 : X = 128
    : A$ = "NRM" : GOTO 430

```

```

320 IF A$ = CF$ THEN A = 0 : X = 64 : A$ =
    "FLS" : GOTO 430

```

```

330 IF A$ = CL$ THEN A = 160 : X = 128
    : A$ = "LWR" : GOTO 430

```

```

340 IF A$ = CC$ THEN A = 64 : X = 128 : A$
    = "CTL" : GOTO 430

```

```

350 IF A$ = C$ THEN 440

```

```

360 IF A$ = CH$ AND C = 0 THEN 270

```

```

370 IF A$ = CH$ THEN C = C - W : GOTO 270

```

```

380 IF C = 30 THEN 270

```

```

390 IF A$ = CUS$ THEN C = C + W : GOTO 270

```

```

400 IF A$ <"@" THEN A$ = CHR$(ASC
    (A$) + X) : GOTO 420

```

```

410 A$ = CHR$(ASC (A$) + A)

```

```

420 PRINT A$ : POKE 540 + C, ASC (A$)
    : C = C + W : GOTO 270

```

```

430 B = PEEK (36) : POKE 56157, 128 :
    HTAB W : PRINT A$ : : POKE 36, B :
    POKE 56157, 0 : GOTO 270

```

```

440 IF C = 0 THEN A$ = C$ : GOTO 470

```

```

450 A$ = "" : FOR A = 0 TO C - W : A$ = A$ +
    CHR$(PEEK (540 + A)) : NEXT

```

```

460 FOR A = W TO 30 - LEN (A$) : A$ = A$
    + CHR$(160) : NEXT : A$ = LEFT$(
    A$, 30)

```

```

470 PR# 0 : CALL 1002 : POKE 56157
    , 128 : RETURN

```

```

480 REM LOCK/UNLOCK SUBROUTINE

```

```

490 F$(CF) = LEFT$(F$(CF), W) + X2$
    + RIGHT$(F$(CF), 38) : RETURN

```

```

500 REM PURGE SUBROUTINE

```

```

510 IF CF = MX THEN 530

```

```

520 FOR A = CF + W TO MX : F$(A - W) =
    F$(A) : NEXT

```

```

530 MX = MX - W : POKE 252, MX + W : IF
    MX = -W THEN A$ = "DISK"
    CONTAINS NO FILES" : GOTO 2410

```

```

540 IF CF > MX THEN CF = MX

```

```

550 RETURN

```

```

560 REM RENAME SUBROUTINE

```

```

570 VTAB 21 : HTAB 16 : PRINT "NEW" :
    : A$ = MIDS (F$(CF), 9, 30) :
    GOSUB 240 : IF A$ = ESC$ THEN POP :
    POP : GOTO 1530

```



```

580 X = 2 : IF LEFT$ (F$(CF) , W) =
  CHR$ (173) THEN X = 3 : AS = LEFT$
  (AS , 29)
590 F$(CF) = LEFT$ (F$(CF) , 8) + AS
  + RIGHT$ (F$(CF) , X) : HOME : X =
  USR (CF) : RETURN
600 REM DELETE SUBROUTINE
610 X = W : IF LEFT$ (F$(CF) , W) =
  CHR$ (173) THEN 680
620 IF RIGHT$ (F$(CF) , 2) = CHR$ (17
  ) + CHR$ (W) THEN 670
630 GOSUB 1150
640 CALL 2453 : FOR A = 32780 TO 33022
  STEP 2 : POKE 254 , PEEK (A) :
  POKE 255 , PEEK (A + W)
650 IF PEEK (A) > 34 OR PEEK (A + W) >
  15 THEN GOSUB 670 : GOTO 1190
660 IF PEEK (A) > 0 THEN CALL 2453 :
  NEXT : GOSUB 1260 : GOTO 640
670 F$(CF) = CHR$ (173) + MIDS
  (F$(CF) , 2 , 36) + MIDS (F$(CF)
  , 39 , W) + CHR$ (255) + RIGHT$
  (F$(CF) , W)
680 VTAB 21 : HTAB W : CALL 64578 :
  RETURN
690 REM EXHUME SUBROUTINE
700 X = 0 : AS = F$(CF) : IF LEFT$ (AS
  , W) <> CHR$ (173) THEN 680
710 F$(CF) = CHR$ (160) + MIDS (AS , 2
  , 36) + CHR$ (160) + MIDS (AS , 38
  , W) + RIGHT$ (AS , W)
720 IF RIGHT$ (F$(CF) , 2) = CHR$ (17
  ) + CHR$ (W) THEN 680
730 GOSUB 1150
740 CALL 2438 : IF PEEK (253) THEN
  F$(CF) = AS : X = W : GOTO 1280
750 CALL 2463 : FOR A = 32780 TO 33022
  STEP 2 : POKE 254 , PEEK (A) :
  POKE 255 , PEEK (A + W)
760 IF PEEK (A) > 34 OR PEEK (A + W) >
  15 THEN 1190
770 IF PEEK (A) = 0 THEN 680
780 CALL 2438 : IF PEEK (253) THEN
  F$(CF) = AS : GOTO 1280
790 CALL 2463 : NEXT : GOSUB 1260 :
  GOTO 740
COR
800 REM SIMULATED CATALOG
810 X = 3 : TEXT : HOME : TEXT : PRINT
  "DISK^VOLUME^" V : PRINT
820 FOR A = 0 TO MX
830 IF LEFT$ (F$(A) , W) = CHR$ (173)
  THEN 860
840 POKE 56157 , 0 : PRINT MIDS (F$(A)
  , 2 , 37) : POKE 56157 , 128 :
  PRINT
850 X = X + W : IF X = 21 THEN X = 0 : GET
  AS
860 NEXT : PRINT
870 PRINT "PRESS^ANY^KEY^TO^
  CONTINUE"
880 WAIT - 16384 , 128 : GET AS : POP :
  GOTO 1530
890 REM GET AN INPUT STRING
900 BS = ""
910 GET AS : IF (AS < A1$ OR AS > A2$)
  AND AS <> CH$ AND AS <> ESC$ AND
  AS <> CM$ THEN 910
920 IF AS = CH$ AND BS = "" THEN 910
930 IF AS = CH$ AND LEN (BS) = W THEN
  PRINT AS : GOTO 900

```

```

940 IF AS = CH$ THEN BS = LEFT$ (BS ,
  LEN (BS) - W) : PRINT AS : GOTO
  910
950 IF AS = CM$ OR AS = ESC$ THEN
  RETURN
960 IF LEN (BS) = LN THEN 910
970 BS = BS + AS : PRINT AS : GOTO 910
980 REM ALLOCATION MAP
990 TEXT : HOME : PRINT SPC (17)
  "TRACK" : PRINT SPC (3) "0" SPC (
  15) "1" SPC (15) "2"
1000 PRINT SPC (3)
  "0123456789ABCDEF0123456789A
  BCDEF012"
1010 AS = "SECTO" : FOR A = 0 TO 4 :
  PRINT "^^" A : NEXT : FOR A = 5 TO 9
  : PRINT MIDS (AS , A - 4 , W) :
  1020 PRINT "^^" A : NEXT : PRINT "R^A"
  : FOR A = W TO 5 : PRINT "^^" CHR$
  (65 + A) : NEXT
1030 FOR A = 0 TO 15 : VTAB A + 4 : HTAB
  4 : POKE 255 , A : FOR X = 0 TO 34 :
  POKE 254 , X : CALL 2438
1040 IF PEEK (253) THEN INVERSE
1050 PRINT "^^" : NORMAL : NEXT : IF
  PEEK (- 16384) <> 155 THEN NEXT
1060 PRINT : PRINT : PRINT
  "HIGHLIGHTED^SECTORS^ARE^
  USED" : PRINT : GOTO 870
1070 REM CURRENT FILE?
1080 VTAB 21 : HTAB W : CALL 64578 :
  PRINT NS (A)
1090 PRINT "F^ENTER^FILENAME , ^C^A
  CURRENT^FILE" : PRINT : PRINT
  "WHICH^C" CH$ :
1100 GET AS : IF AS <> ESC$ AND AS <>
  CM$ AND AS <> "C" AND AS <> "F"
  THEN 1100
1110 IF AS = ESC$ THEN POP : VTAB 21 :
  HTAB W : CALL 64578 : RETURN
1120 IF AS = CM$ OR AS = "C" THEN AS =
  "C"
1130 PRINT AS : RETURN
1140 REM READ IN TRACK SECTOR LIST
1150 POKE 47081 , S * 16 : POKE 47082
  , D : POKE 47083 , 0
1160 POKE 47084 , ASC (MIDS (F$(CF)
  , 39 , W) ) : POKE 47085 , ASC (
  MIDS (F$(CF) , 40 , W) )
1170 POKE 47088 , 0 : POKE 47089 , 128 :
  POKE 47092 , W : POKE 254 , PEEK
  (47084) : POKE 255 , PEEK (47085
  ) : CALL 2105 : RETURN
1180 REM ILLEGAL TRK SECTOR LIST
1190 AS = "ILLEGAL^TRACK^
  SECTOR^LIST"
1200 VTAB 21 : HTAB W : CALL 64578 :
  PRINT AS CHR$ (7) CHR$ (7)
1210 FOR B = 1 TO 1000 : NEXT : IF X
  THEN RETURN
1220 FOR B = 32780 TO A - 2 STEP 2 :
  POKE 254 , PEEK (A) : POKE 255 ,
  PEEK (A + W)
1230 CALL 2438 : CALL 2463 : IF PEEK
  (253) THEN CALL 2453
1240 NEXT : RETURN
1250 REM NEXT TRK SECTOR LIST
1260 POKE 47084 , PEEK (32769) : POKE
  47085 , PEEK (32770) : GOTO 1170
1270 REM FILE NOT EXHUMABLE

```

```

1280 AS = "FILE^NOT^AVAILABLE" :
  GOTO 1200
1290 REM INITIALIZE VARIABLES
1300 CH$ = CHR$ (8) : CD$ = CHR$ (4)
  : CM$ = CHR$ (13) : ESC$ = CHR$ (27
  ) : CU$ = CHR$ (21)
1310 PR# 0 : IN# 0 : CALL 1002 : CI$ =
  CHR$ (9) : CN$ = CHR$ (14) : CF$ =
  CHR$ (6) : CL$ = CHR$ (12) : CC$ =
  CHR$ (3)
1320 W = 1 : TEXT : X = PEEK (49283) +
  PEEK (49283) : POKE 10 , 76 : POKE
  11 , 17 : POKE 12 , 9
1330 KS = CH$ + CU$ +
  " , ^ACDELMPRTU?/" + ESC$ : SPS
  = "" : FOR A = W TO 30 : SPS = SPS +
  CHR$ (160) : NEXT
1340 DIM FS (104) , NS (19) : FOR A = W
  TO 19 : READ NS (A) : NEXT : WILD$ =
  CHR$ (189)
1350 TS = CHR$ (160) + CHR$ (160) +
  CHR$ (212) + CHR$ (160) + CHR$
  (176) + CHR$ (176) + CHR$ (176)
  + CHR$ (160)
1360 FOR A = 768 TO 777 : READ X : POKE
  A , X : NEXT : ONERR GOTO 2660
1370 REM READ DIRECTORY
1380 GOSUB 2800 : VTAB 22 : INVERSE :
  PRINT "INSERT^DISK" : NORMAL
1390 VTAB 22 : HTAB 19 : PRINT
  "SLOT^6" : HTAB 18 : PRINT
  "DRIVE^1" : HTAB 17 : PRINT
  "VOLUME^0" :
1400 LN = W : A1$ = "1" : A2$ = "7" :
  VTAB 22 : HTAB 24 : GOSUB 900 : IF
  AS = ESC$ THEN HOME : POKE 103 , W :
  POKE 104 , 8 : END
1410 IF BS = "" THEN BS = CHR$ ( PEEK (
  PEEK (40) + PEEK (41) * 256 + 23)
  - 128)
1420 S = VAL (BS) : POKE 46529 , S :
  VTAB 23 : HTAB 24 : A2$ = "2" :
  GOSUB 900
1430 IF AS = ESC$ THEN HTAB 24 : PRINT
  W : GOTO 1400
1440 IF BS = "" THEN BS = CHR$ ( PEEK (
  PEEK (40) + PEEK (41) * 256 + 23)
  - 128)
1450 D = VAL (BS) : POKE 46528 , D :
  VTAB 24 : HTAB 24 : A1$ = "0" : A2$ =
  "9" : LN = 3
1460 GOSUB 900 : IF AS = ESC$ THEN
  HTAB 24 : PRINT "0^^" : GOTO
  1400
1470 IF VAL (BS) > 255 THEN HTAB 24 :
  PRINT "0^^" : HTAB 24 : GOTO
  1460
1480 HTAB 24 : PRINT VAL (BS) "^^" :
  POKE 46527 , VAL (BS)
1490 POKE 54 , 90 : POKE 55 , 8 : CALL
  1002 : POKE 252 , 0
1500 FS (0) = "" : CALL 2194 : X = FRE
  (2) : PR# 0 : CALL 1002 : V = PEEK
  (47094)
1510 IF PEEK (252) = W THEN HOME :
  VTAB 12 : PRINT "DISK^CONTAINS^
  NO^FILES" : END
1520 MX = PEEK (252) - 2 : POKE 252
  , MX + W : CF = INT (MX / 2)
1530 TEXT : HOME

```

```

1540 HTAB W : X = USR ( CF )
1550 VTAB 21 : HTAB W : PRINT
      "A,C,D,E,L,M,P,R,T,U,<,>,
      ESC,?=HELP"
1560 VTAB 23 : HTAB W : PRINT
      "COMMAND=>"; : GET AS
1570 FOR A = W TO LEN ( K$ ) : IF AS <>
      MIDS ( K$ , A , W ) THEN NEXT : GOTO
      1560
1580 ON A GOSUB 120 , 140 , 120 , 140
      , 120 , 140 , 990 , 810 , 2190 , 2300
      , 1800 , 2010 , 1680 , 1610 , 1950
      , 1930 , 1590 , 1590 , 2400 : GOTO
      1540
1590 TEXT : GOSUB 2800 : VTAB 23 :
      GOTO 870
1600 REM RENAME FILES
1610 GOSUB 1080 : IF AS <> "C" THEN
      1630
1620 GOSUB 570 : RETURN
1630 GOSUB 230 : B$ = AS : IF AS = ESC$
      THEN POP : GOTO 1530
1640 GOSUB 170
1650 IF X = 0 THEN RETURN
1660 GOSUB 570 : GOSUB 200 : GOTO 1650
1670 REM PURGE FILES
1680 GOSUB 1080 : IF AS <> "C" THEN
      1710
1690 IF LEFT$ ( F$ ( CF ) , W ) = CHR$
      ( 173 ) THEN GOSUB 510
1700 POP : GOTO 1530
1710 GOSUB 230 : B$ = AS : IF AS = ESC$
      THEN 1830
1720 GOSUB 170
1730 IF X = 0 THEN RETURN
1740 IF LEFT$ ( F$ ( CF ) , W ) <> CHR$
      ( 173 ) THEN 1780
1750 VTAB 23 : HTAB W : PRINT
      "PURGE?AY" CH$; : GET AS : IF AS <
      > "Y" AND AS <> "N" AND AS <> CMS
      AND AS <> ESC$ THEN 1750
1760 IF AS = ESC$ THEN RETURN
1770 PRINT AS; : IF AS = "Y" OR AS =
      CMS THEN GOSUB 510 : HOME : GOSUB
      190 : GOTO 1730
1780 GOSUB 200 : GOTO 1730
1790 REM LOCK FILES
1800 X1$ = CHR$ ( 160 ) : X2$ = CHR$
      ( 170 ) : X3$ = "LOCK"
1810 GOSUB 1080 : IF AS <> "C" THEN
      1840
1820 IF MIDS ( F$ ( CF ) , 2 , W ) = X1$
      THEN GOSUB 490
1830 POP : GOTO 1530
1840 GOSUB 230 : B$ = AS : IF AS = ESC$
      THEN 1830
1850 GOSUB 170
1860 IF X = 0 THEN RETURN
1870 IF MIDS ( F$ ( CF ) , 2 , W ) <> X1$
      THEN 1910
1880 VTAB 23 : HTAB W : PRINT X3$ "YA
      Y" CH$; : GET AS : IF AS <> "Y" AND
      AS <> "N" AND AS <> CMS AND AS <>
      ESC$ THEN 1880
1890 IF AS = ESC$ THEN RETURN
1900 IF AS = "Y" OR AS = CMS THEN GOSUB
      490
1910 GOSUB 200 : GOTO 1860
1920 REM UNLOCK FILES

```

```

1930 X1$ = CHR$ ( 170 ) : X2$ = CHR$
      ( 160 ) : X3$ = "UNLOCK" : GOTO
      1810
1940 REM TITLE FILE
1950 VTAB 21 : HTAB W : CALL 64578 :
      PRINT N$ ( A ) : GOSUB 230
1960 IF AS = ESC$ THEN HTAB W : CALL
      64578 : RETURN
1970 POKE 252 , MX + 2 : MX = MX + W
1980 FOR A = MX TO CF + W STEP - W : F$ ( A
      ) = F$ ( A - W ) : NEXT : F$ ( CF ) = T$
      + AS + CHR$ ( 17 ) + CHR$ ( W )
1990 GOTO 2090
2000 REM MOVE FILE
2010 GOSUB 1080 : IF AS = "C" THEN
      2090
2020 GOSUB 230 : B$ = AS : IF AS = ESC$
      THEN RETURN
2030 GOSUB 170
2040 IF X = 0 THEN RETURN
2050 VTAB 21 : HTAB W : CALL 64578 :
      PRINT "MOVE?AY" CH$;
2060 GET AS : IF AS <> "Y" AND AS <>
      "N" AND AS <> CMS AND AS <> ESC$
      THEN 2060
2070 IF AS = ESC$ THEN RETURN
2080 IF AS = "N" THEN GOSUB 200 : GOTO
      2040
2090 VTAB 21 : HTAB W : CALL 64578 :
      PRINT "MOVING^FILE"
2100 HTAB W : X = USR ( CF ) : WAIT -
      16384 , 128 : GET AS
2110 FOR A = W TO 6 : IF AS <> MIDS ( K$
      , A , W ) THEN NEXT : IF AS <> ESC$
      AND AS <> CMS THEN 2100
2120 IF AS = ESC$ OR AS = CMS THEN
      RETURN
2130 A = A - INT ( A / 2 ) * 2 : ON A + W
      GOSUB 2160 , 2140 : GOTO 2100
2140 IF CF = Z THEN RETURN
2150 AS = F$ ( CF ) : F$ ( CF ) = F$ ( CF - W
      ) : F$ ( CF - W ) = AS : GOTO 120
2160 IF CF = MX THEN RETURN
2170 AS = F$ ( CF ) : F$ ( CF ) = F$ ( CF + W
      ) : F$ ( CF + W ) = AS : GOTO 140
2180 REM DELETE FILE(S)
2190 GOSUB 1080 : IF AS = "C" THEN 610
2200 GOSUB 230 : B$ = AS : GOSUB 170
2210 IF X = 0 THEN 2280
2220 IF LEFT$ ( F$ ( CF ) , W ) = CHR$
      ( 173 ) THEN 2270
2230 VTAB 21 : HTAB W : CALL 64578 :
      PRINT "DELETE?AY" CH$;
2240 GET AS : IF AS <> "Y" AND AS <>
      "N" AND AS <> CMS AND AS <> ESC$
      THEN 2240
2250 IF AS = ESC$ THEN RETURN
2260 IF AS = "Y" OR AS = CMS THEN GOSUB
      610
2270 GOSUB 200 : GOTO 2210
2280 VTAB 21 : HTAB W : CALL 64578 :
      RETURN
2290 REM EXHUME FILE(S)
2300 GOSUB 1080 : IF AS = "C" THEN 700
2310 GOSUB 230 : B$ = AS : GOSUB 170
2320 IF X = 0 THEN 2280
2330 IF LEFT$ ( F$ ( CF ) , W ) <> CHR$
      ( 173 ) THEN 2380
2340 VTAB 21 : HTAB W : CALL 64578 :
      PRINT "EXHUME?AY" CH$;

```

```

2350 GET AS : IF AS <> "Y" AND AS <>
      "N" AND AS <> CMS AND AS <> ESC$
      THEN 2350
2360 IF AS = ESC$ THEN RETURN
2370 IF AS = "Y" OR AS = CMS THEN GOSUB
      700
2380 GOSUB 200 : GOTO 2320
2390 REM SAVE DIRECTORY
2400 AS = ""
2410 HOME : PRINT "THE^LONE^
      CATALOG^ARRANGER" : PRINT SPC (
      27 ) "BY^RAY^DARRAH" ;
2420 PRINT : HTAB 20 - LEN ( AS ) / 2 :
      INVERSE : PRINT AS : NORMAL
2430 VTAB 8 : PRINT "S^SAVE^
      DIRECTORY^ON^DISK" : PRINT :
      PRINT "R^RETURN^FOR^
      MORE^EDITING"
2440 PRINT : PRINT "ESC^ARESTART^
      PROGRAM" : PRINT : PRINT
      "WHICH?AS" CH$;
2450 GET AS : IF AS <> "S" AND AS <>
      "R" AND AS <> ESC$ AND AS <> CMS
      THEN 2450
2460 PRINT AS : IF AS = ESC$ THEN RUN
2470 IF AS = "R" AND MX = - W THEN AS =
      "DISK^CONTAINS^NO^FILES" :
      GOSUB 1200 : GOTO 2410
2480 IF AS = "R" THEN HOME : POP : GOTO
      1530
2490 REM BUILD DIRECTORY
2500 HOME : VTAB 21 : PRINT
      "BUILDING^DIRECTORY" : CALL
      2499 : B = 14
2510 FOR A = 33025 TO 36353 STEP 256 :
      POKE A , 17 : POKE A + W , B : B = B - W
      : NEXT : IF MX = - W THEN 2610
2520 CF = 0 : FOR B = 0 TO 15 : FOR A = 11
      TO 221 STEP 35 : AD = 33024 + B *
      256 + A
2530 POKE AD , ASC ( RIGHT$ ( F$ ( CF )
      , 2 ) ) : POKE AD + W , ASC ( RIGHT$
      ( F$ ( CF ) , W ) ) : X = USR ( CF )
2540 AS = MIDS ( F$ ( CF ) , 3 , W ) : FOR X
      = 0 TO 6 : IF AS <> CHR$ ( PEEK ( X +
      45991 ) ) THEN NEXT
2550 X = INT ( ( 2 ^ X ) / 2 ) : IF MIDS
      ( F$ ( CF ) , 2 , W ) = CHR$ ( 170 ) THEN
      X = X + 128
2560 POKE AD + 2 , X : FOR X = 3 TO 32 :
      POKE AD + X , ASC ( MIDS ( F$ ( CF )
      , X + 6 , W ) ) : NEXT
2570 X = ( ASC ( MIDS ( F$ ( CF ) , 5 , W ) )
      - 176 ) * 100 + ( ASC ( MIDS ( F$ ( CF
      ) , 6 , W ) ) - 176 ) * 10
2580 X = X + ASC ( MIDS ( F$ ( CF ) , 7 , W )
      ) - 176 : POKE AD + 33 , X - INT ( X /
      256 ) * 256
2590 POKE AD + 34 , X / 256 : CF = CF + W :
      IF CF <= MX THEN NEXT : NEXT
2600 CF = INT ( CF / 2 ) : HOME : X = USR
      ( CF ) : VTAB 21 : PRINT "ONE^
      MOMENT^PLEASE."
2610 POKE 47081 , S * 16 : POKE 47082
      , D : POKE 47083 , 0 : POKE 47092 , 2
      : POKE 47084 , 17
2620 FOR A = 15 TO W STEP - 1 : POKE
      47085 , A : POKE 47089 , 144 - A :
      POKE 47088 , 0

```



```

2630 CALL 2105 : NEXT : POKE 47089
,179 : POKE 47088 ,187 : POKE
47085 ,0 : CALL 2105
2640 AS = "DIRECTORY"
TRANSFORMATION^COMPLETE" :
GOTO 2410
2650 REM ERROR HANDLER
2660 CALL 768 : ERR = PEEK (222) : IF
ERR > 254 THEN RESUME
2670 IF ERR = 0 OR ERR > 15 THEN HOME :
VTAB 12 : PRINT "ERROR^#^" ERR "A
IN^LINE^" PEEK (218) + PEEK (219
) * 256 ; CHR$ (7) : END
2680 HOME : PRINT "THE^LONE^
CATALOG^ARRANGER" : PRINT SPC (
27) "BY^RAY^DARRAH" : PRINT :
PRINT
2690 AS = "VOLUME^MISMATCH" : IF ERR
= W THEN AS = "WRITE^PROTECTED"
2700 IF ERR = 4 THEN AS = "DRIVE^
I/O^ERROR"
2710 HTAB 20 - LEN (AS) / 2 : INVERSE
: PRINT AS : NORMAL : PRINT :
PRINT CHR$ (7)
2720 PRINT "T>^TRY^AGAIN" : PRINT :
PRINT "R>^RESTART^PROGRAM" :
PRINT : PRINT "ESC>^ABORT^
FUNCTION"
2730 PRINT : PRINT "WHICH^?^T^" CHR$ ;
2740 GET AS : IF AS <> "T" AND AS <>
"R" AND AS <> ESC$ AND AS <> CMS
THEN 2740
2750 PRINT AS : IF AS = "R" THEN RUN
2760 IF AS = ESC$ AND MX > - W THEN
CALL 54915 : GOTO 1530
2770 IF AS = ESC$ THEN AS = "DISK^
CONTAINS^NO^FILES" : GOTO 2410
2780 HOME : VTAB 12 : PRINT "ONE^
MOMENT^PLEASE." : RESUME
2790 REM PRINT HELP SCREEN
2800 HOME : PRINT "THE^LONE^
CATALOG^ARRANGER" : PRINT SPC (
27) "BY^RAY^DARRAH" ;
2810 POKE 32 , 2 : POKE 33 , 38 : PRINT :
PRINT "A>^ALLOCATION^MAP"
2820 PRINT "C>^DISPLAY^A^
SIMULATED^CATALOG" : PRINT :
PRINT "D>^DELETE^FILES"
2830 PRINT "E>^EXHUME^ (BRING^
BACK)^DELETE^FILES" : PRINT :
PRINT "L>^LOCK^FILES"
2840 PRINT "M>^MOVE^FILES^WITHIN^
THE^DIRECTORY" : PRINT : PRINT
"P>^PURGE^DELETED^FILES"
2850 PRINT "R>^RENAME^FILES" :
PRINT : PRINT "T>^CREATE^A^
TITLE^FILE"
2860 PRINT "U>^UNLOCK^FILES" :
PRINT : PRINT "?>^THIS^
HELP^SCREEN"
2870 PRINT "ESC>^STOP^CURRENT^
FUNCTION^ (GO^BACK)" : TEXT :
RETURN
2880 DATA , , , , , , , DELETE , EXHUME
, LOCK , MOVE , PURGE , RENAME
, TITLE , UNLOCK , , ,
2890 DATA 104 , 168 , 104 , 166 , 223
, 154 , 72 , 152 , 72 , 96

```

### The Lone Catalog Arranger Checksums

```

10 - $BADD 740 - $2E2E
20 - $9B13 750 - $CFFE
30 - $4D3B 760 - $0009
40 - $AD92 770 - $6DC2
50 - $C899 780 - $E196
60 - $FF65 790 - $F1C6
70 - $A3BF 800 - $6333
80 - $A900 810 - $D943
90 - $924D 820 - $1E5E
100 - $6ECE 830 - $8DE4
110 - $8C78 840 - $25FF
120 - $69DE 850 - $DA02
130 - $576C 860 - $84DB
140 - $CA57 870 - $E99C
150 - $B273 880 - $CA53
160 - $1D55 890 - $1257
170 - $A90C 900 - $BB77
180 - $0A1D 910 - $F27F
190 - $2248 920 - $388D
200 - $DB6B 930 - $7081
210 - $A60B 940 - $FF6E
220 - $D527 950 - $6CA5
230 - $8935 960 - $1A27
240 - $C49B 970 - $0DE2
250 - $A8C0 980 - $E7E5
260 - $D80B 990 - $2DD9
270 - $29AC 1000 - $3E5A
280 - $9668 1010 - $91EE
290 - $710D 1020 - $61BD
300 - $5F23 1030 - $AE1C
310 - $329D 1040 - $3688
320 - $A575 1050 - $1ACA
330 - $E2E0 1060 - $A426
340 - $D73E 1070 - $9EEE
350 - $9CEC 1080 - $9255
360 - $B627 1090 - $78A1
370 - $D0E4 1100 - $78DF
380 - $B69D 1110 - $3C03
390 - $7A7A 1120 - $1152
400 - $162D 1130 - $337B
410 - $3708 1140 - $6706
420 - $4E49 1150 - $0BC9
430 - $C1DD 1160 - $FD40
440 - $C690 1170 - $B48E
450 - $4F41 1180 - $2EB3
460 - $8D6A 1190 - $631A
470 - $AC71 1200 - $2FC7
480 - $E596 1210 - $96E5
490 - $3AC4 1220 - $0428
500 - $6410 1230 - $B76A
510 - $A200 1240 - $4E60
520 - $FB33 1250 - $A0DE
530 - $B173 1260 - $E617
540 - $E8B2 1270 - $1B1A
550 - $1629 1280 - $AB9A
560 - $4ABF 1290 - $38FE
570 - $2080 1300 - $547D
580 - $7238 1310 - $88E6
590 - $662A 1320 - $D47F
600 - $65ED 1330 - $1314
610 - $4C1F 1340 - $07CC
620 - $E1CE 1350 - $77EF
630 - $3AAE 1360 - $8644
640 - $9FE9 1370 - $B653
650 - $3E85 1380 - $90F4
660 - $00A0 1390 - $E3F0
670 - $39BC 1400 - $542C
680 - $7FBD 1410 - $F114
690 - $CE1F 1420 - $A88F
700 - $0186 1430 - $6985
710 - $0E77 1440 - $9D6D
720 - $08ED 1450 - $42AB
730 - $6907

```

```

1460 - $EFEE 2220 - $D2C7
1470 - $378D 2230 - $ECCA
1480 - $ACD6 2240 - $DEE1
1490 - $8B18 2250 - $A1C3
1500 - $7ADE 2260 - $F21A
1510 - $5301 2270 - $D6D0
1520 - $2E68 2280 - $8872
1530 - $775D 2290 - $9400
1540 - $0A40 2300 - $8C72
1550 - $2F85 2310 - $E272
1560 - $9D0D 2320 - $6812
1570 - $F5C3 2330 - $6ED9
1580 - $437E 2340 - $979C
1590 - $2EA0 2350 - $40CE
1600 - $EDA9 2360 - $B7BA
1610 - $8F36 2370 - $EA2D
1620 - $5808 2380 - $6CE0
1630 - $738A 2390 - $FC4B
1640 - $FDC6 2400 - $4524
1650 - $9B54 2410 - $7DDF
1660 - $6E77 2420 - $A3E9
1670 - $7EAB 2430 - $EC24
1680 - $A6D8 2440 - $1914
1690 - $01FA 2450 - $4E3
1700 - $460C 2460 - $E4EA
1710 - $F336 2470 - $3DB9
1720 - $6D9A 2480 - $E8D7
1730 - $D3C8 2490 - $871F
1740 - $B805 2500 - $6F59
1750 - $6F78 2510 - $22B9
1760 - $65E5 2520 - $2F39
1770 - $4302 2530 - $5054
1780 - $1764 2540 - $0013
1790 - $09DF 2550 - $4FFE
1800 - $9D8A 2560 - $00C0
1810 - $15DB 2570 - $A286
1820 - $D4DF 2580 - $E194
1830 - $5809 2590 - $6FD
1840 - $3247 2600 - $C5DE
1850 - $3A37 2610 - $68CB
1860 - $56D7 2620 - $ABC4
1870 - $79A1 2630 - $529F
1880 - $4891 2640 - $9489
1890 - $C932 2650 - $C69C
1900 - $4006 2660 - $8730
1910 - $5D1E 2670 - $D86B
1920 - $318B 2680 - $180A
1930 - $D5B6 2690 - $39B4
1940 - $2951 2700 - $92D4
1950 - $34A4 2710 - $9FED
1960 - $2154 2720 - $9CD3
1970 - $0849 2730 - $A65D
1980 - $4C61 2740 - $074E
1990 - $0C46 2750 - $D45D
2000 - $C1A6 2760 - $3F8A
2010 - $5D9F 2770 - $C3F0
2020 - $CCE9 2780 - $508C
2030 - $7269 2790 - $0764
2040 - $A0E9 2800 - $F57C
2050 - $1888 2810 - $9824
2060 - $191C 2820 - $ECB0
2070 - $4452 2830 - $4C09
2080 - $7E2E 2840 - $4D98
2090 - $39E9 2850 - $0DDC
2100 - $3053 2860 - $8D93
2110 - $5886 2870 - $D037
2120 - $B834 2880 - $793B
2130 - $6A0F 2890 - $374E
2140 - $2CBD
2150 - $FE39
2160 - $ED38
2170 - $37A1
2180 - $2B32
2190 - $418B
2200 - $D3F1
2210 - $D551

```

CORE



Sideways  
Funk Software  
P.O. Box 1290  
Cambridge, MA 02238  
\$60.00

**Requirements:**

Apple II+ or equivalent  
DOS 3.3 System Master  
One blank disk

**S**ideways by Funk Software is a new and useful utility for the Apple II family of computers. It can be used, with VisiCalc and Multiplan, to print spreadsheets rotated 90 degrees (SIDEWAYS to the way they normally would be printed.) This option is useful for obtaining hardcopy of wide tables which, when printed sideways, will fit on a single 8 1/2 x 11-inch sheet of paper. It can also be used to print very wide tables (such as annual or multi-year projections) on a number of attached sheets of continuous feed paper.

### Backing-Up Sideways

A backup of Sideways is supplied on the reverse side of the program disk. Unfortunately, if the disk is damaged, both the original and the backup could be ruined. Because of this potential problem, I decided to make my own backup of the Sideways program disk.

Normally, I use a nibble copier (EDD III) to make backups. However, I was having trouble copying Sideways without parameters. When I began to investigate the copy protection on the Sideways disk, I found that the disk could be catalogued and, with the exception of Track \$22, the disk has a standard DOS 3.3 format. Because Sideways must use something close to standard DOS, I decided it would be a good candidate for a softkey.

### The Softkey

For those just interested in making a backup of Sideways, the softkey follows. I'll explain later how I developed the softkey.

1) Boot your DOS 3.3 System Master and type

#### RUN COPYA

Make a copy of the original Sideways disk. When you run COPYA, everything will work fine for 4 read cycles. On the 5th read the drive will recalibrate and the program will display the error message: "UNABLE TO READ". Don't worry about this. As I mentioned earlier, Track \$22 is non-standard and a part of the copy protection you'll be circumventing.

2) Put your System Master back in drive 1

# Softkey For SIDEWAYS

By Jack Burke

and type

#### BRUN MASTER CREATE,D1

Following the prompts, give the "greeting" program a filename of "HELLO".

3) Place your COPYA version of Sideways in drive 1 and allow Master Create to turn it into a master disk.

4) Exit MASTER CREATE with the COPYA version still in drive 1. When the disk reboots it will display the error message: "FILE NOT FOUND" since the disk does not yet have a greeting program named HELLO.

5) Type in this short Applesoft program:

```
10 PRINT CHR$(4); "BLOAD^  
SIDEBREAK"  
20 PRINT CHR$(4); "BRUN^SIDEWAYS"
```

6) Save it to your COPYA version by typing  
**SAVE HELLO**

7) Enter the monitor by typing  
**CALL -151**

8) Enter the following hexdump:

```
0300: A0 4C 8C 10 08 A0 12 8C $CAC3  
0308: 1E 08 A0 03 8C 1F 08 4C $2965  
0310: 04 08 A0 EE 8C C3 3E A0 $7573  
0318: 10 8C C8 3E 20 B1 3E 4C $4880  
0320: 20 08 $A5B8
```

9) Save this program to your COPYA version of Sideways

**BSAVE SIDEBREAK,A\$300,L\$22**

10) Load the Sideways program

**BLOAD SIDEWAYS**

11) Modify Sideways to jump to the SIDEBREAK program

**984E:00 03**

12) Save this modified Sideways program back to the COPYA version

**BSAVE SIDEWAYS,A\$9000,L\$EF**

13) You're all done. You can check your copy by typing

**C600G**

(assuming your drive controller is in Slot 6).

### Development Of The Softkey

As I indicated earlier, my first examination of the Sideways program disk revealed that, except for Track \$22, the disk uses standard DOS 3.3 format. My nibble copier would copy Track \$22 but, when I tried to run the copy, it would go part way through a normal boot process and then reboot. This led me to suspect that the program used some form of nibble count as protection.

My first step in tracing the copy protection was to use the Armonitor program (from Hardcore COMPUTIST No. 12) to see which tracks on the disk were actually being used. Loading Armonitor and BRUNing Sideways, I found that: 1) Sideways reads data from tracks which are not in any of the catalogued files and, 2) just before completing the program load, the drive arm seeks Track \$22. With this information in hand I BLOADED Sideways and used BFINd (from Beagle Brothers Utility City) to determine that the Sideways program is loaded in memory starting at \$9000 and has a length of \$EF.

I then began to trace the code starting at \$9000 and determined that program execution would transfer to \$0804 with both the original and nibble copy version of Sideways. After examining the code starting at \$0804, I found at \$825 a JMP \$1151 that, if executed, caused the disk to reboot upon the failure of the nibble count process. This indicated that the copy protection must be in one of the previously executed routines. Tracing back from \$0825, I found JSRs to \$3E03 and \$3E00. I examined the code starting at \$3E00 and found several interesting things. If I interrupted execution of the program before the JSR to \$3E03, a disassembly of the code starting at \$3E11 did not produce a meaningful listing. But if I stopped execution just after the JSR to \$3E03, a disassembly at \$3E11 did produce valid code. *AH HA!*

The code starting at 3E03 is:

```
3E03- A0 CC LDY #SCC  
3E05- 89 10 3E LDA $3E10,Y  
3E08- 49 A7 EOR #SA7  
3E0A- 99 10 3E STA $3E10,Y  
3E0D- 88 DEY
```



## Most Wanted List

If you have been trying to backup a program, and have only ended up pulling your hair out as a result of the ordeal, let us know about it.

We have received softkeys for a number of programs previously in our list and these will be published as soon as each has been evaluated and edited by our staff.

**Hardcore COMPUTIST  
Wanted List  
PO Box 110846-K  
Tacoma, WA 98411**

If you know how to de-protect, unlock or modify any of the programs below, we encourage you to help other Hardcore COMPUTIST readers and earn some extra money at the same time. Send the information to us in article form on a DOS 3.3 diskette.

1. **Apple Business Graphics**  
*Apple Computer*
2. **Flight Simulator II**  
*Sub Logic*
3. **Critical Reading**  
*Borg-Warner*
4. **DB Master 4.0**  
*Stoneware, Inc.*
5. **Bookends**  
*Sensible Software*
6. **Visiblend**  
*Micro Lab*
7. **Dollars And Sense**  
*Monogram*
8. **Lifesaver**  
*Micro Lab*
9. **Catalyst**  
*Quark, Inc.*
10. **Gutenberg Jr. & Sr.**  
*Micromation LTD*
11. **Prime Plotter**  
*Primesoft Corp.*
12. **SSI Wargame Series**  
*Strategic Simulations, Inc.*
13. **Sargon III**  
*Hayden*
14. **Zardax**  
*Computer Solutions*
15. **List Handler**  
*Silicon Valley Systems*
16. **Milliken Math Series (NEW)**  
*Milliken Publishing*

3E0E- D0 F5 BNE \$3E05

Examination of this routine shows that, starting at \$3EDC and stepping backward to \$3E11, each value in memory is EORed with #A7 and then stored back in the same location. The Sideways program uses this technique to disguise the code stored between \$3E11 and \$3EDC until the subroutine at \$3E03 is executed! With my suspicions aroused, I carefully examined the code starting at \$3E11. I found that the first section of code (from \$3E11 to \$3E2C) uses the DOS IOB and RWTS to seek Track \$22. Next, from \$3E2F to \$3EAE, the program uses timing loops to read data from Track \$22 and, depending on the data read, stores specific values in memory locations \$3EC3 and \$3EC8. Then, starting at location \$3EB1, the program uses the values just stored at \$3EC3 and \$3EC8 in a loop similar to the one starting at \$3E03. The loop at \$3EB1 changes the values in memory from \$1800 to \$3EBB. If the values in \$3EC3 and \$3EC8 are not correct, the result will be meaningless code and, even if the program does not reboot at the JMP to \$1151, the program won't be able to continue execution. This means that the routine starting at \$3EB1 is actually using the results of data read from track \$22 (in a custom read routine) to generate the rest of the Sideways code.

After I determined what had occurred, it was a relatively simple matter to stop execution of my original program disk at \$3EB1 and determine the correct values for \$3EC3 and \$3EC8 (BE and 1D, respectively). I then wrote the following short assembly language program named SIDEBREAK.

```
0300- A0 4C LDY #54C
0302- 8C 1D 08 STY $081D
0305- A0 12 LDY #512
0307- 8C 1E 08 STY $081E
030A- A0 03 LDY #503
030C- 8C 1F 08 STY $081F
030F- 4C 04 08 JMP $0804
0312- A0 EE LDY #5EE
```

```
0314- 8C C3 3E STY $3EC3
0317- A0 1D LDY #51D
0319- 8C C8 3E STY $3EC8
031C- 20 B1 3E JSR $3EB1
031F- 4C 20 08 JMP $0820
```

Jumping to this program from \$904E in Sideways allowed me to run the "code creation" routines at \$3E03 and \$3EB1 (with the correct values stored in \$3EC3 and \$3EC8) while avoiding the routine which seeks and subsequently reads data from Track \$22.

### The End

All that remained was to set up a COPYABLE version of Sideways which would 1) load my SIDEBREAK program before the Sideways program began execution, and 2) transfer control to SIDEBREAK from Sideways at the appropriate point. After examining the Armonitor dump I collected earlier and the VTOC on the disk, I determined that I could save small programs directly without overwriting the used but uncatalogued tracks on the disk. I saved my assembly program to disk and then wrote an Applesoft program to first load SIDEBREAK and subsequently run Sideways (Step 5). I then renamed the greeting program (Steps 2 to 4) and made a slight modification to the Sideways program so it would jump to SIDEBREAK (Step 11). The result is a COPYABLE version of a useful utility program.

One last thing that I discovered during my investigation of the Sideways disk should be of interest to Hardcore COMPUTIST readers. It seems that the original Sideways disk contains several deleted files, one of which (TEXT.FUNK) is the source code for the nibble count routine. To discover more details about the workings of the Sideways protection scheme, you may want to UNDELETE this file with a utility such as COPY ][ Plus.

### Bugs In Hardcore COMPUTIST No.'s 13 & 14

#### Hardcore COMPUTIST No. 13, Pg. 16

**CSaver:** Several readers have reported problems using the CSAVER program from Issue No. 13. We have found that the problems have been caused by the use of a NEW command instead of an FP when clearing Applesoft programs. Always use FP, and not NEW, when CSAVER is installed.

#### Hardcore COMPUTIST No. 14, Pg. 22

**Tracking Down Rocky's Boots:** The source code for the patch that is applied to track \$1, sector \$7 of the copy of Rocky's Boots has an incorrect origin. The origin should be at \$4AE and not \$4EA. The instructions contained in the text for making the patch are correct.

Still typing in programs  
that appear in Hardcore COMPUTIST?

## Order the LIBRARY DISK, instead!

Save yourself time and trouble.

Each month a Library Disk with all the programs that appeared in the previous issue of Hardcore COMPUTIST is prepared for SMART READERS like you who have better things to do with their time than type in program listings.

**Disk No. 15** for Hardcore COMPUTIST 15 includes Master Type Controller & Boot From Drive 2.

**Disk No. 14** for Hardcore COMPUTIST 14 includes SAT Controller, Sea Dragon Controller, Super IOB v1.2 with Standard Controller and Swap Controller, Batman Decoder Ring & Rocky's Boots Controller.

**Disk No. 10** for Hardcore COMPUTIST 13 includes Penguin Controller, Snooper Troops Controller, CSaver, New DOS 3.3 Command Listings & Electronic Arts Controller

**Disk No. 9** for Hardcore COMPUTIST 12 includes The Armonitor, Lion's Share Controller, CORE Disk Searcher & Psychedelic Symphony

**Disk No. 8** for Hardcore COMPUTIST 11 includes Ultimaker, Ultimapper, Ultima III Controller & Sensible Speller

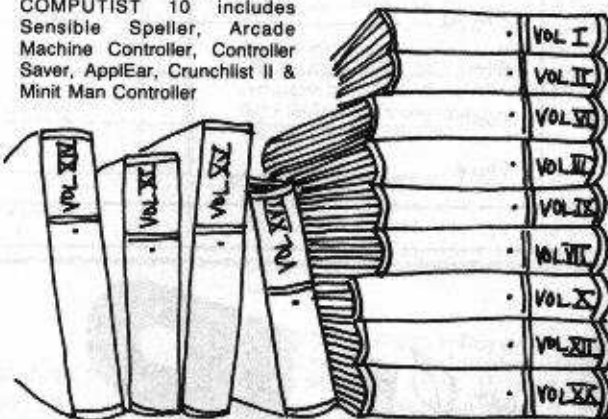
**Disk No. 7** for Hardcore COMPUTIST 10 includes Sensible Speller, Arcade Machine Controller, Controller Saver, ApplEar, Crunchlist II & Minit Man Controller

**Disk No. 3** for CORE Games issue includes Destructive Forces & Dragon Dungeon

**Disk No. 2** includes programs for CORE Utilities issue and Hardcore COMPUTIST 4

**Disk No. 1** includes programs for CORE Graphics issue and Hardcore COMPUTIST 1

Get each  
Library Disk  
for the low price  
of  
**\$9.<sup>95</sup>**



(Don't miss  
the **SPECIAL**  
Library Disk & Back  
Issue combination  
offers on page 29)

NOTE: Library Disks need accompanying magazine for documentation. Library Disks for sold-out issues are available upon request.

Send check or money order to:  
Hardcore COMPUTIST, PO Box  
110846-B, Tacoma, WA 98411.  
Most orders shipped UPS. Please  
use street address. Washington  
residents add 7.8% sales tax.  
Foreign orders add 20% shipping  
and handling. US funds drawn on  
US bank.

(If you're interested in receiving  
the accompanying Library Disk  
each month automatically with  
your magazine subscription, just  
write for details.)

Please send me the Library Disk(s) I have indicated below. I have enclosed \$9.95 (plus applicable tax & shipping) for each disk.

- Disk No. 15
- Disk No. 14
- Disk No. 10
- Disk No. 9
- Disk No. 8
- Disk No. 7
- Disk No. 3
- Disk No. 2
- Disk No. 1

### SPECIAL!

Order FIVE Library Disks  
of your choice  
for only \$40.00!

No. \_ / \_ / \_ / \_ / \_

Name \_\_\_\_\_ ID# \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_

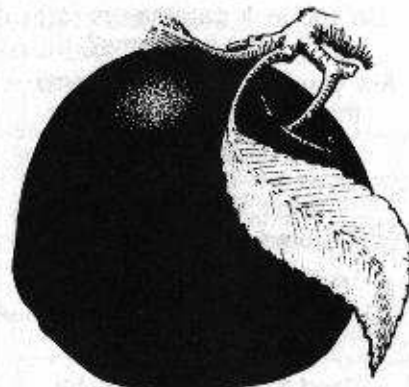
Phone \_\_\_\_\_

VISA/MC \_\_\_\_\_ Exp \_\_\_\_\_

Signature \_\_\_\_\_

# NOT PIRACY

Just Good Sense!



## SUBSCRIBE to Hardcore COMPUTIST and...

- Make backups more easily
- Move software from floppy to hard disk
- Add custom modifications such as fast-DOS to speedup LOADs and SAVEs

### Annual Subscription Rates:

Please check one

- U.S. .... \$25
- Canada, U.S. 1st Class ..... \$34
- Mexico ..... \$39
- Foreign Airmail ..... \$60
- Foreign surface mail ..... \$40
- SAMPLE, U.S. .... \$3.50
- SAMPLE, Foreign ..... \$4.50

( ) Yes, start my subscription now.  
( ) I would like to RENEW my subscription. (Please paste PRESENT MAILING LABEL below)

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_  
Phone \_\_\_\_\_  
VISA/MC \_\_\_\_\_ Exp \_\_\_\_\_  
Signature \_\_\_\_\_

Send check or money order (US funds drawn on US bank) to: Hardcore COMPUTIST, PO Box 110846-T, Tacoma, WA 98411



## Apple II, II+, IIe, Franklin users:

Do you have problems  
backing-up your  
copy-protected programs?

Do you lack parameters for  
your copy programs?

Are you looking for programs  
that you can AFFORD?

Are you hesitating to  
upgrade your equipment due to  
expensive prices  
quoted in other ads?

**It's simple now.  
Just drop us a line.**

Send \$1.00 U.S. funds to:

Reliant  
P.O. Box 33610  
Sheungwan, Hong Kong

**IMPORTANT:** We have over 600 PC  
name-brand programs and various  
hardware offers. Programs @ U.S.  
\$8.00/PC includes the disk and  
registered airmail handling.

## Moving Soon?

Let us know your new address **at least 30 days  
in advance** so you won't miss a single issue of  
Hardcore COMPUTIST. Fill out this form, paste  
your present address label in the space provided,  
and send it to:

Hardcore COMPUTIST, Subscription Department  
PO Box 110846-T, Tacoma, WA 98411

My new address is:

Address \_\_\_\_\_

City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_

Phone \_\_\_\_\_

Paste present address label here or fill in old address:

Name \_\_\_\_\_ ID # \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_

Country \_\_\_\_\_

Phone \_\_\_\_\_

# BACK UP YOUR DISKS

**NOW  
AVAILABLE  
AT YOUR LOCAL  
COMPUTER STORE**

## ESSENTIAL DATA DUPLICATOR III™

EDD runs on Apple II, II plus, IIe, IIc and Apple III (in emulation mode) using one or two disk drives.

EDD allows you to easily and quickly make back up copies of your "uncopyable" Apple disks. ■ Since EDD has been preset to copy the widest range of copy-protections possible, you just simply boot up EDD, put the disk you want to copy in one disk drive and a blank disk in the other (EDD will work using one drive also) and in about 2 ½ minutes a copy is made. ■ Unlike the "copy-cards" which only copy "single load" programs, EDD copies the entire disk. This would be similar to hooking up two cassette recorders, playing from one, and recording to the other. ■ We have even included an option so you can check the speed of your disk drives because drive speeds running fast or slow can damage disks and cause other problems.

■ We publish EDD program lists (information about copy-protected disks) every couple of months, which EDD owners can receive. The current list is included with the purchase of EDD. ■ The bottom line is this; if EDD can't copy it, chances are nothing will.

**\$79.95** Ask for EDD at your local computer store, or, to order direct, send \$79.95 plus \$2 shipping (\$5 foreign). Mastercard/Visa accepted. Prepayment required.

**UTILICO MICROWARE**  
3377 Solano Ave., Suite #352  
Napa, CA 94558 (707) 257-2420

**Warning:** EDD is  
sold for the sole  
purpose of  
making archival  
copies ONLY.

Looking for a  
**SPECIAL  
 DEAL**  
 on Library Disks &  
 Back Issues from  
**Hardcore  
 COMPUTIST?**

Check out these  
**SPECIAL  
 Combination  
 OFFERS!**

CORE 1, 2, 3 ..... \$8.50

Hardcore COMPUTIST  
 1, CORE 1, & Library  
 Disk #1 ..... \$14.95

Hardcore COMPUTIST  
 4, CORE 2, & Library  
 Disk #2 ..... \$14.95

Hardcore COMPUTIST  
 1, 4, 10-15, CORE 1, 2,  
 3, & Library Disks #1,  
 2, 7-10, 14, 15 (Limited  
 supplies-orders filled  
 as received) ..... \$105.00

Hardcore COMPUTIST  
 1, 4, 10-15 (Limited  
 supplies) ..... \$26.00

Please send the SPECIALS I have checked above to:

Name \_\_\_\_\_ ID# \_\_\_\_\_  
 Address \_\_\_\_\_  
 City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_  
 Phone \_\_\_\_\_  
 VISA/MC \_\_\_\_\_ Exp \_\_\_\_\_  
 Signature \_\_\_\_\_

Send check or money order to: Hardcore COMPUTIST, PO Box 110846-B, Tacoma, WA 98411. Orders shipped UPS. Please use your street address. Washington residents add 7.8% sales tax. Foreign orders add 20% shipping and handling. US funds drawn on US bank. See the ad on pg. 27 for a complete list of library disks.

**Are you a  
 NEW SUBSCRIBER?**

BACK ISSUES of  
 Hardcore COMPUTIST and ★ CORE are  
 PACKED with information that you  
 won't want to miss.

(See the SPECIAL back issue and library disk offer on this page, too!)

**Hardcore COMPUTIST 15:** Softkeys for Mastertype, Stickybear BOP, Tic Tac Show, The Financial Cookbook, Escape From Rungistan, Alien Munchies, Millionaire & Plato / MREAD/MWRT Update / Cumulative Index to Hardcore Publications: 1981-1984 / A Boot From Drive 2 / DB Master's Data Compression Techniques

**Hardcore COMPUTIST 14:** Super IOB v1.2 Update / Putting Locksmith 5.0 Fast Copy Into a Normal Binary File / Softkeys for Seadragon, Rocky's Boots, Knoware, PFS Software, Computer Preparation SAT & MatheMagic / Batman Decoder Ring / REVIEW: Boulder Dash and A Fix For DiskEdit.

**Hardcore COMPUTIST 13:** Softkeys for Laf Pak, Beyond Castle Wolfenstein, Transylvania and The Quest, Electronic Arts, Snooper Troops (Case 2), DLM Software, Learning With Leeper, & TellStar / CSaver: The Advanced Way to Store Super IOB Controllers / Adding New Commands to DOS 3.3 / Fixing A ProDOS 1.0.1 BSAVE Bug / REVIEW: Enhancing Your Apple / Locksmith 5.0 and the Locksmith Programming Language.

**Hardcore COMPUTIST 12:** Softkeys for Zoom Grafix, Flip Out, Lion's Share, Music Construction Set, Hi-Res Computer Golf II, Suicide, Sabotage, Millionaire, Time Is Money, & Type Attack, Psychedelic Symphony / The CORE Disk Searcher / The Armonitor / Pseudo-ROMs on the Franklin Ace.

**Hardcore COMPUTIST 11:** Copy II Plus 4.4C Update / PARMS for Essential Data Duplicator / Ultimaker III / Mapping of Ultima III / Ultima II...The Rest of the Picture / Softkeys for Sensible Speller, Ultima III, Softporn Adventure, The Einstein Compiler v5.3, & Mask of the Sun.

**Hardcore COMPUTIST 10:** Controller Saver / Softkeys for The Arcade Machine, Bankstreet Writer, Mint Man, Sensible Speller IV, Essential Data Duplicator III, Krell LOGO, & Canyon Climber, ApplEar / REVIEWS: The 65SC802 & 65SC816 Chips and Dino Eggs / Examining Protected Applesoft Basic Programs / Crunchlist II / Parms for DB Master v4.

**Hardcore COMPUTIST 4:** Ultima II Character Editor / Softkeys for Ultima II, Witness, Prisoner II, & Pest Patrol / Adventure Tips for Ultima II & III / Copy II Plus PARMS Update.

**Hardcore COMPUTIST 1:** Softkeys for Data Reporter, Multiplan & Zork / PARMS for Copy II Plus / No More Bugs / APT's for Choplifter & Cannonball Blitz / Reviews: Replay, Crackshot, Snapshot & Wildcard copy cards.

**CORE 3 Games:** Constructing Your Own Joystick / Compiling Games / GAME REVIEWS: Over 30 of the latest and best / Pick Of The Pack: All-time TOP 20 games / Destructive Forces / EAMON / Graphics Magician and GrafORTH / and Dragon Dungeon.

**CORE 2 Utilities:** Dynamic Menu / High Res: Scroll Demo / GOTO Label: Replace / Line Find / Quick Copy: Copy.

**CORE 1 Graphics:** Memory Map / Text Graphics: Marquee, Boxes, Jagged Scroller / Low Res: Color Character Chart / High Res: Screen Cruncher, The UFO Factory / Color / Vector Graphics: Shimmering Shapes, A Shape Table Mini-Editor / Block Graphics: Arcade Quality Graphics for BASIC Programmers / Animation.

(\* CORE is no longer published as an independent quarterly magazine. Back issues not listed are no longer available.)

Send me the back issues indicated below:

HC 15 ..... \$3.50 <input type="checkbox"/>	HC 10 ..... \$3.50 <input type="checkbox"/>
HC 14 ..... \$3.50 <input type="checkbox"/>	HC 4 ..... \$3.50 <input type="checkbox"/>
HC 13 ..... \$3.50 <input type="checkbox"/>	HC 1 ..... \$3.50 <input type="checkbox"/>
HC 12 ..... \$3.50 <input type="checkbox"/>	CORE 1 Graphics ..... \$3.50 <input type="checkbox"/>
HC 11 ..... \$3.50 <input type="checkbox"/>	CORE 2 Utilities ..... \$3.50 <input type="checkbox"/>
	CORE 3 Games ..... \$3.50 <input type="checkbox"/>

Name \_\_\_\_\_ ID# \_\_\_\_\_  
 Address \_\_\_\_\_  
 City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_  
 Phone \_\_\_\_\_  
 VISA/MC \_\_\_\_\_ Exp \_\_\_\_\_  
 Signature \_\_\_\_\_

Send check or money order to: Hardcore COMPUTIST, PO Box 110846-B, Tacoma, WA 98411. Most orders shipped UPS. Please use street address. Washington residents add 7.8% sales tax. Foreign orders add 20% shipping and handling. US funds drawn on US bank.

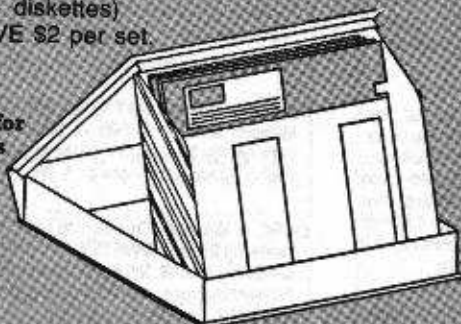


# Add COLOR to your computer library!

Order 10 diskettes for the low price of \$17.00 AND get one color-coded library case absolutely FREE!

Order three or more sets (30+ diskettes) and SAVE \$2 per set.

No shipping charges for US orders



SS/DD 5 1/4" diskettes with hub rings. Includes Tyvek sleeves. Guaranteed, 100% certified, full surface tested. (Matching labels accompany each color coded library case.)

Please send me:

\_\_\_\_\_ set(s) 10 SS/DD diskettes with 1 FREE color coder library case per set

1 or 2 sets @ \$17.00 each \$ \_\_\_\_\_

3 or more sets @ \$15.00 each \$ \_\_\_\_\_

Specify color choices:

Applicable taxes or shipping \$ \_\_\_\_\_

\_\_\_\_\_ yellow \_\_\_\_\_ red \_\_\_\_\_ green TOTAL \$ \_\_\_\_\_

\_\_\_\_\_ beige \_\_\_\_\_ blue \_\_\_\_\_ black \_\_\_\_\_ grey

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_

Phone \_\_\_\_\_

VISA/MC \_\_\_\_\_ Exp \_\_\_\_\_

Signature \_\_\_\_\_

Send order to: SoftKey Publishing, PO Box 110816, Tacoma, WA 98411. Washington residents add 7.8% sales tax. Foreign orders add 20% shipping and handling. US funds drawn on US bank. For faster service, send postal money order or certified check. Orders shipped via UPS. Please use street address.

# SIMPLY SOFTWARE

DISCOUNT SPECIALS ON APPLE SOFTWARE • ALL PRICES 30% OFF

LIST \$24.95 Survival Adventure	SPECIAL \$17.47 Typing Tutor Weather Fronts	LIST \$49.95 Alexander the Great Clickart Crossword Magic Deadline The Eating Machine Enchanter Flight Simulator II Home Cataloger Math Blaster! Planetfall Print Shop	SPECIAL \$34.97 Question Run for the Money Sargon III Sorcerer Spelling Machine Suspended Typing Tutor III Witness Wizardry Word Attack! Zoom Graphix	LIST \$20.00 Apple Alarm	SPECIAL \$14.00
LIST \$29.95 Alphabet Zoo Early Games for Y.C Kings Testing Ground	SPECIAL \$20.97 Poovan Stone of Sisyphus Threshold	LIST \$59.95 Millionaire	SPECIAL \$41.97 S.A.M. Ultima III	LIST \$49.50 Apple Grade Book	SPECIAL \$34.65
LIST \$34.95 BC's Quest for Tires Beyond Castle Wolfenstein Centipede Donkey Kong Early Elementary I Frogger HI-RES Computer Golf	SPECIAL \$24.47 Kamakaze Lions' Share Lode Runner Night Mission Pinball Pacman Story Machine Titan Empire Transylvania	LIST \$69.95 Bank Street Speller Bank Street Writer	SPECIAL \$48.97 Homework Speed Reader II	LIST \$124.95 Bookends	SPECIAL \$87.47
LIST \$39.95 Algebra 1 Algebra 2 Antonym Antics Broadsides Face Maker Master Type Rendezvous	SPECIAL \$27.97 Seastalker Spellcopter Starcross Triple Dump Type Attack Zaxxon Zork I, II and III	LIST \$100.00 Business Forecasting Module Dental Insurance Form Writer	SPECIAL \$70.00 Dollars & Sense Time is Money	LIST \$395.00 BPI General Ledger	SPECIAL \$276.50
		LIST \$125.00 Chart Koalapid Touch Tablet Megamerge PFS: File	SPECIAL \$87.50 PFS: Graph PFS: Report PFS: Write Sensible Speller	LIST \$350.00 DB Master 4.0	SPECIAL \$245.00
		LIST \$495.00 Broderbund General Ledger w/Payables	SPECIAL \$348.50 dBase II Wordstar	LIST \$35.00 Death in the Carribean	SPECIAL \$24.50
				LIST \$30.00 Disk Recovery	SPECIAL \$21.00
				LIST \$145.00 Grappler+	SPECIAL \$101.50
				LIST \$36.00 Highrise	SPECIAL \$25.20
				LIST \$74.95 Home Accountant	SPECIAL \$52.47
				LIST \$39.00 Knoware	SPECIAL \$27.30
				LIST \$39.00 Mac the Knife	SPECIAL \$27.30
				LIST \$54.95 Mach III Joystick	SPECIAL \$38.47
				LIST \$19.95 Meteoroids in Space	SPECIAL \$13.97
				LIST \$40.00 Micro Cookbook	SPECIAL \$28.00
				LIST \$195.00 Multiplan	SPECIAL \$138.50
				LIST \$49.44 Rocky's Boots	SPECIAL \$34.61
				LIST \$92.50 S-C Macro Assembler	SPECIAL \$64.75
				LIST \$19.95 Sargon II	SPECIAL \$13.97
				LIST \$49.00 SAT Word Attack	SPECIAL \$34.30
				LIST \$60.00 Sideways	SPECIAL \$42.00
				LIST \$44.95 Snooper Troops #1	SPECIAL \$31.47
				LIST \$99.95 Speedstar	SPECIAL \$69.97
				LIST \$99.95 Step by Step	SPECIAL \$69.97
				LIST \$89.95 Study Program for SAT	SPECIAL \$62.97
				LIST \$40.00 Summer Games	SPECIAL \$28.00
				LIST \$89.95 Systems Saver	SPECIAL \$62.97
				LIST \$180.00 Tax Manager	SPECIAL \$126.00
				LIST \$150.00 Think Tank	SPECIAL \$105.00
				LIST \$79.95 Ultraplott	SPECIAL \$55.97
				LIST \$379.00 Ultraterm	SPECIAL \$265.30
				LIST \$250.00 Visicalc	SPECIAL \$175.00
				LIST \$149.00 Visischedule	SPECIAL \$104.30

PLEASE make check or M.O. payable to: Simply Software Inc. • P.O. Box 36068 • Kansas City, Missouri 64111  
Add \$3.00 shipping. Missouri residents add 5 5/8% sales tax. Allow 4-6 weeks for delivery.

Public Domain Software  
**SPECIAL**  
 From The COMPUTER LEARNING CENTER!

GET Over 170  
**BUSINESS  
 & FINANCE**  
 Programs FOR ONLY  
**\$28.<sup>00</sup>**

- Forecast your investment profits, and evaluate the worth of your stock portfolio
- Experiment with Visicalc formulas, real estate and stock plots and design a business or personal letter writing format
- Organize your household expenses
- Balance your checkbook, manage your loans, payroll

The Business & Finance SPECIAL contains the following programs and many more...

Annuly	Real Estate Plot	Savings Growth
Bond Price & Interest	Visicalc Formulas	Treasury Bill Valuation
Budget Monthly	Business Finance	Address File
Decision Matrix	Check Stub	Compound Int. Tables
Financial Pak	Household Exp. Profile	File Manager
Keogh Savings Program	Income Tax 1040 For 77	Home Accounting
Mortgage Calculation	Inventory Company	Names Search
Regular Deposits I	Average Growth Rate	File Cabinet I
Stocks	Check Book Balancer	Family Finance
Calendar Personal	Income Taxes	Gen Ledger Printer
Letter Writer	Interest Earned	Constar
Phone List	Loan Balance	etc...

**ORDER NOW!**

You will also receive a complete catalog of Public Domain Software available from the CLC with your order.

Yes! Please send me the Business & Finance Special Offer (Volumes 18-25 from the CLC Public Domain Library). I understand that Public Domain Software is not commercial quality and is supplied as-is and that orders are filled on double sided disks.

Name \_\_\_\_\_ S-4

Address \_\_\_\_\_

City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_

Country \_\_\_\_\_ Phone \_\_\_\_\_

VISA/MC \_\_\_\_\_ Exp \_\_\_\_\_

Signature \_\_\_\_\_

Send check or money order (US funds drawn on US bank) to:  
 Computer Learning Center, PO Box 110876-HC, Tacoma, WA  
 98411. Washington residents add 7.8% sales tax. Foreign orders  
 add 20% shipping and handling.

*(The Computer Learning Center's PDS Library contains over 175 volumes, all which will run on Apple ][ Plus computers and Apple compatibles. Most will also run on Apple IIe and IIc.)*

WHO  
 LIKES  
 THE  
**CIA** ?



- HERE ARE JUST A FEW OF THE MANY
- "an essential part of the Apple-user's repertoire" - APPLE USER
  - "a valuable buy ... manual is practically worth having on its own" - WASHINGTON APPLE PI
  - "the folks at Golden Delicious should be commended ... worth waiting for" - HARDCORE COMPUTIST
  - "multifaceted" - NIBBLE
  - "the most comprehensive disk accessor I have ever come across" - A.B., VERNON, CANADA
  - "its ability to unlock other programs will greatly help me" - DR. B.P., SAN FRANCISCO, CALIFORNIA
  - "an excellent set of programs ... just great - and good value too" - E.A.S., MILTON KEYNES, ENGLAND
  - "very, very educational ... great manual ... it is FANTASTIC!!!!!!" - J.C., TUCSON, ARIZONA
  - "a very enlightening piece of software/book ... top of my list for good buys" - H.S., BLAINE, MINNESOTA
  - "I like yours the BEST" - R.R., CHICAGO

Why all the excitement about the CIA (confidential information advisors)? Probably because it is the ONLY set of utilities (5 in all) which enable even a beginner to investigate, edit, locate, list, trace, rescue, translate, patch, repair, verify, examine, protect, unprotect, analyse, encrypt, and decrypt programs on normal AND protected disks. You also get the "CIA Files", a 65000+ word book which contains detailed instructions for using the C.I.A. plus easy-to-follow, hand-holding tutorials about patching, repair, formatting, encoding, protection, and numerous other disk topics. You'll find plenty of material here which has never before appeared in print. PROGRAMS NOT COPY PROTECTED.

To put the 5 C.I.A. utilities, plus book, on the trail of your Apple II/IIe (R) disks, send \$65.00 by check or money order to:

GOLDEN DELICIOUS SOFTWARE LTD.  
 350 Fifth Avenue, Suite 3308, Dept H, New York, New York 10118



**Dealer Inquiries  
 Invited**

For information on how to obtain  
 Hardcore COMPUTIST at a substantial  
 savings, write:

Hardcore COMPUTIST, PO Box 110816,  
 Tacoma, WA 98411 or CALL (206) 581-6038



.....An inside look at disk formats -- **DISKVIEW**

.....Deprotecting disks with **SUPER IOB**

.....A quick and easy way to **UNLOCK HYPERSPACE WARS**

.....Taking a peek at **BOOT CODE TRACING**

.....List of Publisher abbreviations and **INTRODUCTION TO 'PARMS'**

.....The Compleat Guide to **LOCKSMITH PARAMETERS**

5.....Step-by-Step guide to making backups using **NIBBLES AWAY II PARAMETERS**

38.....Technical notes and making backups using **BACK-IT-UP II + PARAMETERS**

38.....How to make backups using **COPY II PLUS PARAMETERS!**

46.....Curing those Auto-Start ROM blue **HARDWARE SOLUTIONS**

47.....A MENU HELLO PROG.

51.....USING BOTH SIDES OF YOUR DISK

Advanced Playing Techniques, or how to **net INSIDE CASTLE WOLFENSTEIN**

Understand String... **TEXT INVADERS**

If you took all the old Hardcores...  
 Tore off the fancy covers...  
 Deleted all the editorial material,  
 Out-of-date interviews and letters...  
 Updated the remaining material, and THEN  
 included the **MOST RECENT** and **MOST COMPLETE LIST** of parameters  
 for the major bit-copy programs...  
 And packed it all into a single volume...  
 You'd have the core of Hardcore Computing.

We Call It:

## THE BEST OF HARDCORE COMPUTING

Please send me:

- The Best Of Hardcore Computing AND Program Disk.....\$19.95  
 The Best Of Hardcore Computing.....\$14.95  
 Program disk only.....\$9.50

Name \_\_\_\_\_ ID# \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ St \_\_\_\_\_ Zip \_\_\_\_\_

Phone \_\_\_\_\_

Visa/MC \_\_\_\_\_ Exp \_\_\_\_\_

Signature \_\_\_\_\_

Send check or money order to: Hardcore COMPUTIST, PO Box 110846-B, Tacoma, WA 98411.  
 Washington State residents add 7.8% sales tax. Foreign orders add 20% shipping & handling.  
 US funds drawn on US bank.

*By Hackers  
 For Hackers*

- ELITE BOARD DOWNLOADS
- CRACKING TIPS
- PHREAKING SECTION
- GAME CHEATS
- PARMS
- PROGRAMS
- INTERVIEWS
- ADVENTURE TRIPS
- HACKING TIPS
- MYSTERY SECTIONS

Published on both sides of  
 an Apple diskette -  
 4 times a year.

# The BOOT-LEGGER MAGAZINE

**Subscribe Now!**

Send 25 Bucks for a 1-Year Subscription  
**THE BOOT LEGGER**, 3310 Holland Loop  
 Road, Cave Junction, Oregon 97523.  
 Overseas Subscriptions \$50.  
 Canadian \$30 U.S. Currency.

FOR AD INFO. & QUESTIONS  
 CALL BOOTLEG AT (503) 592-4461

# DISCOUNTS!

## 5 1/4 DISKETTES & STORAGE

- SS/DD, BOX OF 10 ..... \$12.00
- SS/DD, 10 BOXES ..... \$115.00
- DOUBLE-NOTCHED DS/DD, EACH ..... \$1.35
- DOUBLE-NOTCHED DS/DD, 100 ..... \$125.00
- HARD PLASTIC STAND-UP DISKETTE LIBRARY CASES ..... \$2.75 EACH  
4 FOR \$10.00  
(specify color choices: beige, black, blue, green, grey, red, yellow)
- SMOKED PLASTIC JUMBO-SIZE FLIP-TOP 75 DISKETTE FILE CASES ..... \$16.00\*
- 70-DISKETTE FILE CASES ..... \$12.00\*
- 140-DISKETTE LOCKING WOOD FILE CABINET ..... \$33.00

## PRINTERS

- PANASONIC P1090 ..... \$239.00
- PANASONIC 1091 ..... \$299.00
- EPSON RX-80 F/T ..... \$289.00
- EPSON RX-100 ..... \$389.00
- EPSON FX-80 ..... \$379.00
- OKI-DATA MICROLINE 92A DOT MATRIX ..... \$369.00
- SILVER REED 400 LETTER QUALITY ..... \$269.00
- STARWRITER A-10 25CPS LETTER QUALITY ..... \$495.00
- TOSHIBA 1340 DOT MATRIX AND LETTER QUALITY COMBINED ..... \$795.00

## PRINTER INTERFACES AND ACCESSORIES

- STANDARD PARALLEL INTERFACE CARD ..... \$49.00
- GRAPHICS PARALLEL INTERFACE CARD ..... \$75.00
- FINGERPRINT PUSH-BUTTON GRAPHICS CARD ..... \$119.00
- MICROFAZER GENERAL PRINT BUFFER ..... \$149.00
- PRINTER STAND ..... \$14.00
- SWITCH BOX
- 3 PARALLEL PORTS ..... \$129.00
- SWITCH BOX, 3 SERIAL PORTS ..... \$79.00

## FLOPPY DISK DRIVES

- FOURTH DIMENSION (FULL OR SLIMLINE) ..... \$179.00
- DISTAR ..... \$159.00
- LASER ..... \$139.00\*
- IIc DRIVES ..... \$158.00\*
- DISK CONTROLLER ..... \$59.00
- DOUBLE-SIDED DRIVE ..... \$199.00
- 650K RANA DRIVE ..... \$439.00

## HARD DISK DRIVES

- 5 MEGABYTE WITH CONTROLLER AND SOFTWARE ..... \$749.00
- 10 MEGABYTE ..... \$1175.00

## MONITORS

- GORILLA 12-INCH GREEN ..... \$84.00
- USI 12-INCH GREEN ..... \$94.00\*
- USI 12-INCH AMBER ..... \$99.00\*
- INTRA 14-INCH COMPOSITE COLOR/80 COLUMN ..... \$239.00

## MODEMS

- ZOOM TELEPHONICS 300-BAUD ..... \$109.00
- CENTAURI 300 BAUD ..... \$179.00
- PRO-MODEM 1200 ..... \$349.00
- PRO-MODEM 1200A INTERNAL ..... \$279.00
- SINGALMAN MARK XII ..... \$239.00

## GRAPHICS DEVICES

- POWER PAD & STARTER KIT ..... \$99.00

## VIDEO & DISPLAY EQUIPMENT

- DIGITIZER ..... \$299.00
- B & W CAMERA ..... \$195.00
- COLOR PROCESSOR ..... \$99.00
- COLOR PROCESSOR/ENHANCER STABILIZER/SYNTHESIZER ..... \$279.00

## GENERAL ITEMS

- 6 OUTLET POWER STRIP ..... \$19.00
- SURGE PROTECTOR ..... \$11.00
- RF MODULATOR ..... \$49.00
- COMPUTER STAND ..... \$24.00

## GAME I/O DEVICES

- CH MACH II JOYSTICK ..... \$37.00
- CH MACH III JOYSTICK ..... \$45.00

\* DENOTES NEW PRICE OR ITEM

## SLOT EXPANSION

- 16 RAM CARD ..... \$49.00
- 64K RAM & 80 COLUMN CARD FOR IIc ..... \$109.00
- MEMORY MASTER IIc 64K + RAM & 80 COLUMN CARD ..... \$145.00
- MEMORY MASTER IIc 128K RAM & 80 COLUMN CARD ..... \$175.00\*
- MICROTEK II + 128 K VISICALC AND MEMORY EXPANSION ..... \$219.00
- MODEM ELIMINATOR CABLE ..... \$21.00
- SERI-ALL SERIAL INTERFACE CARD ..... \$119.00
- 80-COLUMN CARD (VIEWMASTER) WITH SOFT-SWITCH ..... \$129.00
- CENTAURI APS Z-80 CARD ..... \$59.00
- Z-80 PLUS CARD (CPM FOR APPLE) ..... \$115.00
- FAST Z-80 CARD APPLICARD ..... \$175.00
- TIMEMASTER II CLOCK/CALENDAR CARD ..... \$109.00
- QUICK-LOADER PROM BOARD ..... \$149.00
- ANALOG/DIGITAL BOARD ..... \$99.00
- SUPER I/O BOARD ..... \$49.00
- MULTIPLE-SLOT EXPANSION CHASSIS ..... \$149.00
- SINGLE-SLOT EXTENDER ..... \$29.00

## SPECIAL PERIPHERALS

- COOLING FAN WITH SURGE PROTECTOR ..... \$39.00
- TITAN KEYBOARD ..... \$159.00
- LIFETIME EXTERNAL POWER SUPPLY ..... \$179.00
- SHIFT KEY MOD KIT ..... \$8.00
- SCREEN SWITCHER/ DRIVE STEPPER ..... \$74.00

## APPLE SOFTWARE

- WORD STAR ..... \$195.00
- MAIL MERGE/SPELL STAR/ STAR INDEX ..... \$125.00

LONG DISTANCE  
CALL TOLL FREE  
WITH TOUCH TONE PHONE  
FROM ANY CITY!  
DIAL 950-1088  
WAIT FOR TONE  
DIAL 363-1313



ORDERS &  
CALL BACK  
MESSAGES  
(ANSWERING  
MACHINE)  
(202) 362-9176

VISIT OUR NEW STORE  
LOCATION  
8231 WOODMONT AVE.  
(AT BATTERY LANE)  
IN BETHESDA

## STORE HOURS:

12-8 M-TH  
12-6 FRI  
11-5 SAT

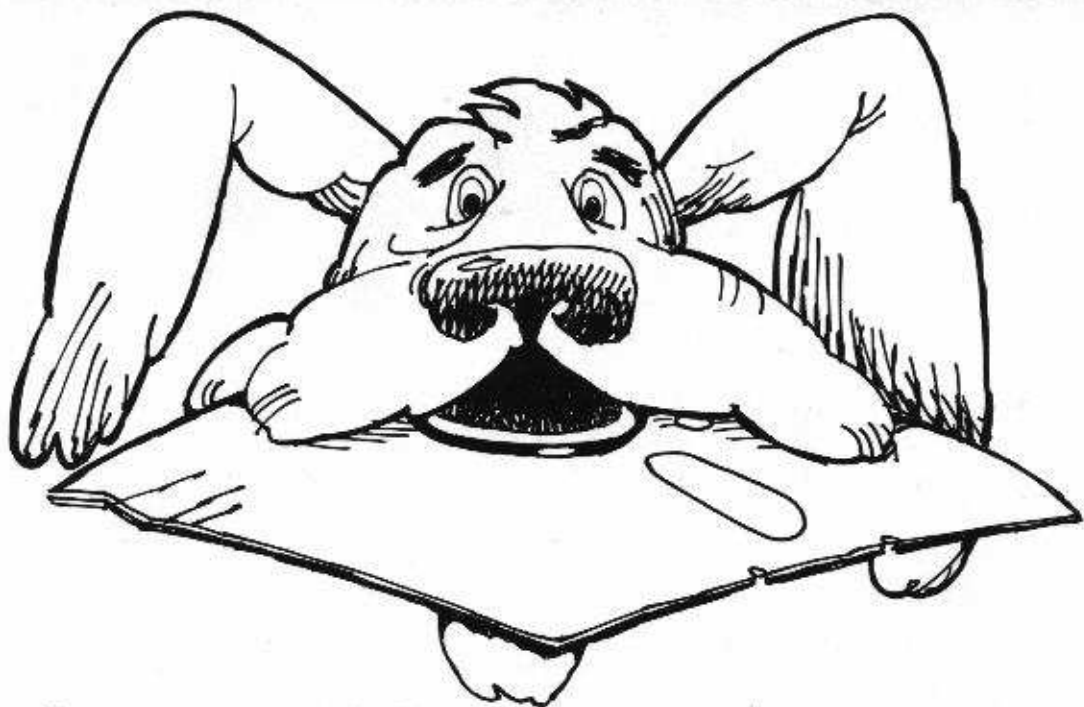
SPECIAL  
APPLE IIc  
PRINTER  
CABLES  
\$19.00\*

UPS shipping,  
\$4.00 per order  
plus \$6.00  
per printer  
or monitor

(202) 363-1313  
**W ASSOCIATES**

8231 WOODMONT AVE., BETHESDA, MARYLAND





## Having problems with retrieval?

### You need software insurance.

Diskettes are fragile, and when a protected program is damaged, the results are expensive and inconvenient. If you have a backup diskette, though, you can have your Apple, IBM or compatible computer back on line within seconds... affordably. That's software insurance.

### Copy II Plus (Apple ][, ][ Plus, //e)

This is the most widely used backup program for the Apple. Rated as "one of the best software buys of the year" by InCider magazine, its simple menu puts nearly every disk command at your fingertips. The manual, with more than 70 pages, describes protection schemes, and our Backup Book™ lists simple instructions for backing up over 300 popular programs. A new version is now available that is easier to use and more powerful than before. Best of all, Copy II Plus is still only \$39.95.

### WildCard 2 (Apple ][, ][ Plus, //e)

Designed by us and produced by Eastside Software, WildCard 2 is the easiest-to-use, most reliable card available. Making backups of your total load software can be as easy as pressing the button, inserting a blank disk and hitting the return key twice. WildCard 2 copies 48K, 64K and 128K software, and, unlike other cards, is always ready to go. No preloading software into the card or special, preformatted diskettes are required. Your backups can be run with or without the card in place and can be transferred to hard disks. \$139.95 complete.

**Important Notice:** These products are provided for the purpose of enabling you to make archival copies only. Under the Copyright Law, you, as the owner of a computer program, are entitled to make a new copy for archival purposes only, and these products will enable you to do so.

These products are supplied for no other purpose and you are not permitted to utilize them for any use, other than that specified.

### Copy II PC (IBM)

This is THE disk backup program for the IBM PC and PC/XT that backs up almost anything. Others may make similar claims, but in reality, nothing out performs Copy II PC... at any price. Copy II PC even includes a disk speed check and is another "best buy" at only \$39.95.

We are the backup professionals. Instead of diluting our efforts in creating a wide variety of programs, we specialize in offering the very best in backup products. So, protect your software investment, before your software meets its master.



## CENTRAL POINT Software, Inc.

The Backup Professionals

To order, call 503/244-5782, 8:00-5:30 Mon.-Fri., or send your order to: Central Point Software, 9700 SW Capitol Hwy, Suite 100, Portland, OR 97219. Prepayment is required. Please include \$3 for shipping and handling (\$8 outside U.S. or Canada).



Welcome