For The Serious User Of Apple][Computers

COMPUTIST

Issue No. 39

January 1987

USA $3.75
Canada/Mexico $7.00
All Others $13.25

Softkeys For:

MIDI/8 PLus
Homeword v2.1
Amazon
Echo Plus
The Luscher Profile
A. E.
Bridge Baron
Borrowed Time

Core:

Repairing Keyboards

Feature:

Electronic Disk
Drive Swapper

(Page 26)

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

Coping With COMPUTIST

Welcome to COMPUTIST, a publication devoted to the serious user of Apple][and Apple][compatible computers. Our magazine contains information you are not likely to find in any of the other major journals dedicated to the Apple market.

New readers are advised to read this page carefully to avoid frustration when attempting to follow a softkey or when entering the programs printed in this issue.

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

■ **Commands And Controls:** In any article appearing in COMPUTIST, commands which a reader is required to perform are set apart by being in boldface and indented:

PR#6

The RETURN key must be pressed at the end of every such command unless otherwise specified.

Control characters are specially boxed:

6CTRLP

Press 6. Next, place one finger on CTRL and press P. Remember to enter this command line by pressing RETURN.

■ **Requirements:** COMPUTIST programs and softkeys require one of the Apple][series of computers and a disk drive with DOS 3.3. These and other special needs are listed at the beginning of the article under "Requirements".

■ **Software Recommendations:**

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

2) Sector Editor such as DiskEdit (from the Book of Softkeys vol I) or ZAP from Bag of Tricks.

3) Disk Search Utility such as The Inspector, The CIA or The CORE Disk Searcher (from the Book of Softkeys vol II).

4) Assembler such as the S-C Assembler from S-C software or Merlin/Big Mac.

5) Bit Copy Program such as Copy][Plus, Locksmith or The Essential Data Duplicator

6) Text Editor (that produces normal sequential text files) such as Applewriter II, Magic Window II or Screenwriter II.

COPYA, FID and MUFFIN from the DOS 3.3 System Master Disk are also useful.

■ **Super IOB:** This powerful deprotection utility (COMPUTIST 32) and its various controllers are used in many softkeys. This utility is now available on each Super IOB Collection disk.

■ **RESET Into The Monitor:** Softkeys occasionally require the user to stop the execution of a copy-protected program and directly enter the Apple's system monitor. Check the following list to see what hardware you will need to obtain this ability.

Apple][Plus - Apple //e - Apple compatibles: 1) Place an Integer BASIC ROM card in one of the Apple slots. 2) Use a non-maskable interrupt (NMI) card such as Replay or Wildcard.

Apple][Plus - Apple compatibles: 1) Install an F8 ROM with a modified RESET vector on the computer's motherboard as detailed in the "Modified ROM's" article (COMPUTIST 6 or Book Of Softkeys III) or the "Dual ROM's" article (COMPUTIST 19).

Apple //e - Apple //c: Install a modified CD ROM on the computer's motherboard. Cutting Edge Ent. (Box 43234 Ren Cen Station-HC; Detroit, MI 48243) sells a hardware device that will give you this important ability but it will void an Apple //c warranty.

■ **Recommended Literature:** The Apple][Reference Manual and DOS 3.3 manual are musts for any serious Apple user. Other helpful books include: *Beneath Apple DOS*, Don Worth and Pieter Lechner, Quality Software; *Assembly Language For The Applesoft Programmer*, Roy Meyers and C.W. Finley, Addison Wesley; and *What's Where In The Apple*, William Lubert, Micro Ink.

■ **Keying In Applesoft Programs:** BASIC programs are printed in COMPUTIST in a format that is designed to minimize errors for readers who key in these programs. If you type:

10HOME:REMCLEAR SCREEN

The LIST will look like:

10 HOME : REM CLEAR SCREEN

because Applesoft inserts spaces into a program listing before and after every command word or mathematical operator. These spaces usually don't pose a problem except in line numbers which contain REM or DATA commands. There are two types of spaces: those that have to be keyed and those that don't. Spaces that must be keyed in appear in COMPUTIST as delta characters (^). All other spaces are there for easier reading. NOTE: If you want your checksums (See "Computing Checksums" section) to match up, you must only key in (^) spaces after DATA statements.

■ **Keying In Hexdumps:** Machine language programs are printed in COMPUTIST as both source code and hexdumps. Hexdumps are the shortest and easiest format to type in. You must first enter the monitor:

CALL -151

Key in the hexdump exactly as it appears in the magazine, ignoring the four-digit checksum at the end of each line (a "$" and four digits). A beep means you have typed something that the monitor didn't understand and must, therefore, retype that line.

When finished, return to BASIC with:

E003G

BSAVE the program with the correct filename, address and length parameters given in the article.

■ **Keying In Source Code** The source code is printed to help explain a program's operation. To key it in, you will need the S-C Assembler. Without this assembler, you will have to translate pieces of the source code into something *your* assembler will understand. A table of S-C Assembler directives appears in COMPUTIST 17.

■ **Computing Checksums** Checksums are four-digit hexadecimal numbers which tell if you keyed a program exactly as it appears in COMPUTIST. There are two types of checksums: one created by the CHECKBIN program (for machine language programs) and the other created by the CHECKSOFT program (for BASIC programs). Both appeared in COMPUTIST 1 and The Best of Hardcore Computing. An update to CHECKSOFT appeared in COMPUTIST 18. If the published checksums do not match those created by your computer, then you typed the program incorrectly. The line where the first checksum differs has an error.

■ **CHECKSOFT Instructions:**

LOAD *filename*
BRUNCHECKSOFT

Get the checksums with: **&** RETURN and correct the program where the checksums differ.

■ **CHECKBIN Instructions:**

CALL -151
BLOAD *program filename*

Install CHECKBIN at an out of the way place

BRUN CHECKBIN,A$6000

Get the checksums by typing the starting address, a period and ending address of the file followed by a CTRLY RETURN.

xxx.xxxCTRLY

Correct the lines at which the checksums differ.

You have a LEGAL RIGHT to an unlocked backup copy

Our editorial policy is that we do NOT condone software piracy, but we do believe that users are entitled to backup 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 programs to meet his or her needs.

Furthermore, the copyright laws guarantee your right to such a DEPROTECTED backup copy:

..."It is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or ***adaptation*** of that computer program provided:

1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or

2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner."

United States Code title 17, §117 (17 USC 117)

You asked for it...

A Disk–Magazine Combo Subscription!

COMPUTIST now has a combination library disk & first-class subscription rate to save you even more. With this new 'COMBO' subscription, you will receive each monthly issue AND its corresponding disk for as much as 43% off the individual rate.

COMBO subscriptions are sent U.S. First Class mail. AND, If you have at least 3 issues left on your current subscription, you can even UPGRADE to this new offer.

OLD NEWS:

COMPUTIST's annual subscription rate has DROPPED.

Old Rates:
*U.S. Domestic **$40** per year*
*U.S. First Class **$48** per year*
*Canada and Mexico **$68** per year*
*All other foreign **$120** per year*

Yes I want to subscribe. Enclosed are U.S. Funds (drawn on a U.S. bank) for a 12 issue subscription.

□ **New Subscriber** □ **Please renew my subscription**

□ **U.S. $32** □ **U.S. First Class/Canada/Mexico $45** □ **Other Foreign $75**

Combination magazine and corresponding disk subscriptions:

□ **U.S./Canada/Mexico $100** □ **Other Foreign $140**

To upgrade your subscription to a combo subscription, U.S./Canada/Mexico send $5.50 per remaining issue. Other Foreign send $6.50 per remaining issue. You must have at least 3 issues remaining to take advantage of this upgrade offer.

Name_________________________ ID#________
Address_________________________
City_____________ State_____ Zip________
Country_____________ Phone____________
VISA ________-________-________ Exp.______
Signature_________________________ CP39

U.S. Funds drawn on U.S. bank. Subscription will not commence until funds are received. Send orders to: COMPUTIST PO Box 110846-T Tacoma, WA 98411

EXPIRED?

Keep your subscription

ALIVE!

Check your mailing label to see if you need to renew your subscription. And if you think you might forget when the time arrives, renew right now. Just use this order blank. Remember, late renewals cannot be back dated.

if you're moving

Let us know at least 30 days in advance so that you won't miss a single issue. Just write your new address on the order form above, and your old address below (or include your present address label).

■ Issues missed due to non-receipt of Change-of-Address may be acquired at the regular back-issue rates. Please remember, the Post Office does not forward third class mail unless requested.

■ COMPUTIST is not responsible for replacing issues lost while forwarding order is in effect.

Name_________________________ ID#________
Address_________________________
City_____________ State_____ Zip________
Country_____________ Phone____________

S.O.S.

(Save On Software)

Title	Publisher	Price	QTY	Total Cost
Recommended Literature:				
□ Beneath Apple DOS (Book)	Quality Software	$16.00	___	______
□ Beneath Apple ProDOS (Book)	Quality Software	$16.00	___	______
Recommended Software:				
□ Global Program Line Editor	Beagle Bros	$35.25	___	______
□ Magic Window // (specify][or //e)	Artsci	$106.00	___	______
□ Bag of Tricks II	Quality Software	$39.75	___	______
Miscellaneous Bargains				
□ Homeword Word Processor & Speller (indicate 64K or 128K)	Sierra On-Line	$55.50	___	______
□ Homeword Filer	Sierra On-Line	$55.50	___	______
□ Borrowed time	Activision, Inc	$32.00	___	______
□ Amazon	Spinnaker Software	$12.75	___	______
□ Speed Reader][	Davidson & Assoc.	$49.50	___	______
□ Donald Ducks Playground	Sierra On-Line	$24.00	___	______
□ Great American Rd Race	Activision, Inc	$32.00	___	______
□ Computer Prep for SAT	Harcourt, Brace	$62.25	___	______
□ Beyond Castle Wolfenstein	Muse Software	$24.50	___	______
□ The Caverns of Frielag	Muse Software	$12.00	___	______
□ Silent Service	Microprose	$27.75	___	______
□ Swashbuckler	Datamost	$15.00	___	______
		Subtotal		______
		Shipping*		$3.00
		Total		______

*Foreign orders, please inquire as to appropriate shipping fees.

Name ______________________________ ID# __________

Address ______________________________

City ______________ State ______ Zip ______________

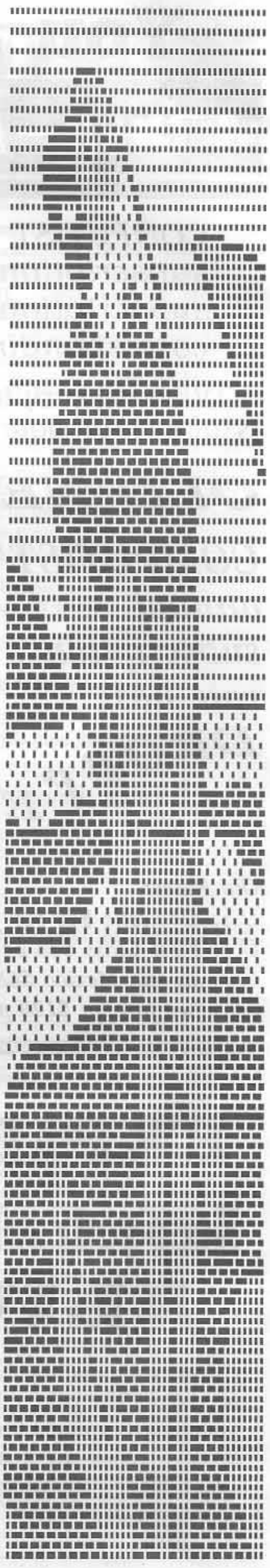
Country ______________ Phone ______________

VISA ______-______-______-______ Exp. __________

Signature ______________________________ CP39

To order, complete order form and mail to:
COMPUTIST PO Box 110937-SOS Tacoma, WA 98411

Offer good while supplies last. Washington residents add 7.8% sales tax. Foreign orders inquire as to appropriate shipping and handling fees. Limited offer, expires February 28, 1986.



Graphics by Spinnaker Software's 'Amazon'

This month's cover:
Graphics from Trillium's "Amazon."

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

Unsolicited manuscripts are assumed to be submitted for publication at our standard rates of payment. SoftKey publishing purchases all and exclusive rights. For more information on submitting manuscripts, consult our writer's guide.

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 COMPUTIST magazine or SoftKey Publishing.

COMPUTIST will replace lost issues for 60 days following the publication date. We cannot be held responsible for mail loss after 60 days.

Apple usually refers to an Apple][computer and is a trademark of Apple Computers, Inc.

SUBSCRIPTIONS: Rates (for 12 issues): U.S. $32, U.S. 1st Class, Canada & Mexico $45, Foreign $75. Direct inquiries to: **COMPUTIST, Subscription Department, PO Box 110846-T, Tacoma, WA 98411.**

DOMESTIC DEALER RATES: Call (206) 474-5750 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.

COMPUTIST

Issue 39 January 1987

Publisher/Editor: Charles R. Haight **Managing Editor:** Ray Darrah
Technical Editor: Robert Knowles **Circulation:** Debbie Holloway, Helen Dornes
Advertising: (206) 474-5750 **Printing:** Valco Graphics Inc., Seattle, WA
COMPUTIST is published monthly by SoftKey Publishing, 5233 S. Washington, Tacoma, WA 98409
Phone: (206) 474-5750

softkeys:

features:

core:

hints:

departments:

input

Please address letters to:

COMPUTIST
Editorial Department
PO Box 110846-K
Tacoma, WA 98411

Include your name, address and phone number.

Correspondence appearing in the INPUT section may be edited for clarity and space requirements. In addition, because of the great number of letters that we receive and the small size of our staff, a response to each letter is not guaranteed.

Our technical staff is available for phone calls between 1:30 pm and 4:30 pm (PST) on Tuesdays and Thursdays only.

Opinions expressed are not necessarily those of COMPUTIST or SoftKey Publishing

A Square Deal

I've got four things to say here, the first is being to compliment you on your excellent magazine, it's well worth the subscription price! The next is to contribute to the already huge pile of **Rescue Raiders** APTs: John Clements detailed all the "ZIPPY" commands in COMPUTIST No. 26, but left out the "shield" command. Pressing <RETURN> toggles an invisible shield which protects your helicopter from all machine gun fire (from other helicopters, men, tanks, AA guns, everything!) and shrapnel. I would also like to add that "ZIPPY" is also a toggle, so typing it twice turns it off. Here are some Rescue Raiders sector edits:

Trk	Sct	Byte	From	To	Effects
ØD	ØØ	64	Ø2	FØ	missiles
ØF	Ø4	9A & 9B	FØ 2D	EA EA	bombs
ØF	Ø4	2D & 2E	FØ E3	EA EA	bullets

The first two give you infinite missiles and bombs. The last one changes your normal machine gun bullets into napalm (and you get hundreds of 'em!). I advise making a backup of your backup because these sector edits do strange things (besides their listed function).

Again, my next subject has been detailed, (COMPUTIST No. 34 "More ROM Running" by Wes Felty), and again I have something to add: a fun thing to do with your language card is to modify ROM (WAITAMINIT! Modify WHAT?), thats right ROM, or at least a copy of ROM. For this trick you must have a language card (standard on a //e and //c):

```
CALL -151
CØ81:Ø
DØØØ<DØØØ.FFFFM
CØ83:Ø
CØ83:Ø
```

Now an active copy of the ROM (in the language card RAM) can be modified, even saved to disk and loaded at another time! To restore the original ROM hit RESET. If that doesn't work, them power down and up again. Notice this also lets you modify Applesoft!

Finally, here is a program to add a textfile viewing command to DOS 3.3 in place to the useless "CHAIN" command:

```
Ø3ØØ- AØ Ø4 B9 2D Ø3 99 92 A8
Ø3Ø8- 88 DØ F7 A9 DE 8D 26 9D
Ø31Ø- A9 BC 8D 27 9D AØ 21 B9
Ø318- 31 Ø3 99 DE BC 88 DØ F7
Ø32Ø- AØ ØØ B9 98 A8 99 97 A8
Ø328- C8 CØ 6F DØ F5 6Ø 56 49
Ø33Ø- 45 D7 2Ø C6 A5 A9 8D 2Ø
Ø338- ED FD 2Ø EF BC DØ F8 4C
Ø34Ø- EA A2 AD ØØ CØ C9 AØ FØ
Ø348- Ø3 4C 8C A6 8D 1Ø CØ A9
Ø35Ø- ØØ 6Ø ØØ ØØ ØØ ØØ ØØ ØØ
```

To save, type:

```
]BSAVE VIEW,A$3ØØ,L$52
```

To install it, type:

```
]BRUN VIEW
```

To use this, first install it, then type "VIEW <filename>" and the text file will be dumped to your screen. <SPACE> will abort the viewing, and ⌘S will pause it. This is a very handy program for a BBS (I should know).

Now you can INIT a disk and the patch will be permanent. Be careful though because other patches may use the same space at $BCDF because it is a large free area of DOS.

Radioactive Snail
Dallas, TX

All Pirates, Please Stand Up

This letter is to support the one in COMPUTIST No. 33.

No one buys original software these days except 13-year-olds and computer user groups. Out of my 6ØØØ disks, I have 8 pieces of original software. All of which, either came with a piece of hardware I bought, or someone gave it to me to crack and let me keep his original (my faithful customers). But the world is wrong about us pirates. We do more than crack software. Pirates have written some great programs! Who do you think makes all of those Music Construction Data Disks? Electronic Arts? Ha! Who do you think wrote programs like Nibbles Away and Copy II Plus? You show me one person in the entire U.S. that has all original software, and I'll show you 1Ø,ØØØ Pirates who will laugh him out of the country! Even Mr. Moses, who claims to hate piracy, but would not have any kind of software library without us. Face it, piracy is such a popular trade, some guys make money for doing it. Me and a bunch of friends own a company called Sectorware Enterprises, and we charge $50 an hour to crack software. We have money coming out of our ears and its all coming from the so-called people who are against pirates.

Mr. Sector
Address Unknown

Fixing Karateka

In reply to Mr. Batchelor's problem with the Karateka softkey (COMPUTIST No. 34), I too had a problem. Somewhere around step 1Ø (maybe earlier), when it did some code reading, the computer would hang, and pressing reset dropped me into the monitor, but it garbaged parts of memory. My solution was to keep the drive running. To accomplish this, one of two ways may be used. First, you can just leave the drive on throughout the entire softkey (opening the drive door will prevent wear and tear on the disk). At no time should you type (from the monitor) CØE8, which shuts the drive moter off. The second way is a little harder, and it requires a little more than 1K of memory in your head to remember. After each step (or whenever there is disk access), type CØE8 (from the monitor). This will shut off the drive motor. Continue the step, but just before the last instruction, type CØE9. This turns the drive back on. Even though you may not need it in some places, it's a good idea to do it, just in case. Right around step 14 this procedure can be scratched. I also used Mr. Wong's Karateka Fix (COMPUTIST No. 3Ø). It stated (for those of you that missed it) that step 17 didn't work (A851G did not reconnect DOS). To get around this, boot the slave disk and save the appropriate parts, then loop back to step 15.

Toney Phalen
Garden Grove, CA

Fixing Crisis Mountain

I have tried a few different softkeys for Crisis Mountain, but Rene Gaudet's softkey in COMPUTIST No. 34 was the first one to work for me. There was only one problem in that if you got a high score the program would bomb! I fixed this by using a sector editor and getting track Ø, sector 2 from a disk with normal DOS. Then I wrote it back to the Cracked Crisis Mountain disk. It worked!!!

Next I tried Danny Pollak's APT for bypassing the title page, which takes forever. It didn't work so I played around with it and made the following changes to track 7 to totally skip the title page.

TRACK	SECTOR	BYTE	FROM	TO
7	ØB	AC	ØØ	18
7	ØB	D3	ØC	3A
7	ØC	EC	2Ø	4C
7	ØC	ED	53	99
7	ØC	EE	49	42
7	ØD	4C	4C	4C
7	ØD	4D	BC	99
7	ØD	4E	I FORGOT!	42

Now the game starts right off with asking you to hit button Ø. I hope this will help some of the high score players out there.

Robert Rowe
Spring Valley, CA

Exploring Bank Street Writer

The Bank Street Writer softkey of COMPUTIST No. 25 has one big bug. If you make the sector edit on track $Ø4, sector $Ø7, byte $37 as directed, it becomes impossible to save documents to the data disk.

If the byte is restored to normal value, documents can be saved but as noted in the article the utilities will not save changes to the disk.

Investigation of track $Ø4, sector $Ø7 shows that it contains the data table to change prologues and epilogues from program disk to data disk values. I changed the program bytes to normal DOS values and sure enough without making the edit to byte $37 the utilities functioned perfectly. However the Data disk could again not store data.

I then changed one byte of the now normal program prologue bytes to $FF (it doesn't matter which byte); the result: utility saves don't function but data disk saves function normally.

Obviously BSW compares the data and program values in the table. If they are both normal some flag is set and data-disk saves produce a write-protect error. This problem could be easily eliminated if I knew one piece of information - At what memory location is the data table stored?

I would then sector edit $Ø4 $Ø7 to produce normal prologues and epilogues except for one byte. I would then sector edit Track $ØA, sector $ØA to replace the words "please wait...." with a poke to the table in memory to insert the one program table byte not yet normal. This would cause the utility program to normalize the program bytes just before the data is saved to the softkeyed disk and yet would allow the word processor to find the program and data disk bytes incompatible as expected.

In an unrelated matter, in a letter of an earlier issue, I read that MECC software softkeys with a swap controller. Recently I softkeyed eight MECC packages and found it necessary to include pokes to do extended retries in the read routine (as per the Newsroom softkey) as well as a poke to ignore unreadable sectors. Without those pokes three of the packages could not be done.

Jim Bancroft
So. Dartmouth, MA

Ann's Interlude with Karateka

Dearest Computist:

Interlude II is an "adult computer game" currently being sold by mail order. The game is protected, but can easily be made COPYAable. The steps to do so are as follows:

1) Use COPYA to copy both sides of the Interlude II disk.

2) Put the copy of the boot side of the Interlude disk into a drive and type the following:

LOAD HELLO
UNLOCK HELLO
25 POKE 36864,96: REM DEFEAT DISK CHECK
SAVE HELLO
LOCK HELLO

Please don't say as you did in issue COMPUTIST No. 34 that the Karateka softkey and most others can't be performed on the //c. I did it fine using the technique outlined on page 6 of COMPUTIST No. 26 in R. G. Henderson's letter on deprotecting Dazzle Draw on the //c. (There is a typo in his step-by-step procedure, however. In step 5 it should read "863C:86".) I also had to incorporate Sam Wong's suggestion in his letter on page 6 of COMPUTIST No. 3Ø. Dazzle Draw too fell to the mighty softkey.

For those brave adventurers among you who want to rescue the fair maiden in your spare time without devoting too many hours, here's a simple cheat for Karateka. It presumes you have a broken copy of Karateka.

To modify the program:

1) Boot DOS

2) BLOAD KARATEKA

3) Type in the following code to point the reset vector to the monitor:

```
2Ø88: A9 59 8D F2 Ø3 A9 FF 8D
2Ø9Ø: F3 Ø3 A9 5A 8D F4 Ø3 4C
2Ø98: ØC Ø2
```

4) BSAVE KARATEKA,A$2ØØØ,L$51ØØ

1) At any time (as long as you haven't been killed) press CONTROL-RESET to get into the monitor. Then type the following to select a segment, enable the language card, and turn graphics back on. Type <RETURN> after each item.

2) ctrl E

3) Type a colon, followed by the number of the segment (1 to 4) you want to jump to. For example:

:3
CØ83
CØ83
CØ57
CØ5Ø

At this point you will have the graphics screen back and won't be able to see what you typed.

F62ØG

As always, Love
Ann Onymous

input

It's Certificate Maker!

Being a very new subscriber to your magazine I just wanted to write and say that it is great. I only wish that I had started my subscription several years ago when you publication first started coming out. I could have saved myself many long hours of work deprotecting some of the programs which I have done.

Upon receiving my first issue, COMPUTIST No. 33, I sat down and read the entire magazine. I enjoyed the softkeys the most but the other areas were very informative. Keep up the great work.

Now on to the real reasons for this letter. First, I have just received the program Certificate Maker, from Springboard, and found it to be ProDOS. As I have a Franklin Ace 12ØØ it annoyed me. The Franklin users in the world must modify ProDOS programs to get them to work. As I did not want to modify the original I decided I would make a backup and modify it. I was not surprised that the backup did not work because the normal disk copy (Copy][Plus) had a read error on track $22, side A. I then set out to find the routing which did the suspected nibble count. Not being very good at keeping good notes on what I am doing, I can not give the full details of all the boot trace work that it required. It is somewhat boring anyway (I am sure that some will disagree with this because they like details) and it is the end result that counts. With the program in memory the code at $7DE1 was 2Ø AC 82 or JSR $82AC. By placing three EAs there the program worked fine. I then used the sector editor of Copy II+ 6.Ø program to find this code on the disk. I found it on track $Ø3 sector $ØØ bytes $71, $72, and $73 on side A of the program disk. After making the EA changes and writing back to the disk everything worked fine. The other sides of this program disk copied just fine using Copy II+ 6.Ø.

Keep up with the great magazine.

William B. Beach
Warminster, PA

Pacman Goodies

In COMPUTIST No. 27, Terry Yancey asked if anybody knew how to get to the "Rank or Cartoon" message in Pacman. I have found out how to get to this option, plus one that allows you to play the game in "Black & White or Color." Here is an easy 4-step way to do it:

1) BLOAD a cracked Pacman program.

2) Get into the monitor and change locations $1144 through 1146 from 4C 17 42 to EA's.

```
CALL -151
1114:EA EA EA
```

3) While in the monitor, also change locations $1821 through $1823 from 4C A6 48 to EA's.

```
1821:EA EA EA
```

4) Now re-save the modified Pacman.

```
BSAVE PAC-MAN, A$1ØØØ, L$6AØØ
```

Now, when you BRUN Pacman, you will see the question: "Black & White or Color? (B/C)." Just press "B" or "C." You should then be at the "Keyboard or Joystick" prompt. Press the usual 1 or 2 to select the control you wish. After hitting the appropriate key, you should be in the "Attract" mode. Press space to exit.

Finally, you should see the "Rank or Cartoon?" message. Rank is the level of play you will start at and cartoon is an option to see the intermission cartoons. To choose one, just type "R" or "C" and return, then press a number key and return.

Glen Talbot
Hampton, IL

Unbootable Copy Cards?

I recently was talking to a friend about copy protection and he mentioned to me something I had never heard before. He said that some companies are using a routine to check for a copycard in your computer and then not boot if it finds one. Even though I do not own one, I thought the idea was rather simple and was surprised someone had not thought of it before. Am I behind the times or have you not heard of this type of protection?

Also, back in COMPUTIST No. 33, I saw the softkey for Thief and was stunned when I saw the graphics on the accompanying pages. The graphics were from "Heist", by Microlab. I have seen the game Thief and the graphics are much different.

Neal Jhala
Concord, CA

Mr. Jhala: While it is true in theory that a routine could check for a copy card, most (about 98%) of them are designed in such a manner that it is impossible to detect their presence.

In checking with our layout department, it appears as if they goofed on the Thief softkey. The graphics are indeed from Heist. Oh Well!

Most Wanted List

Need help backing-up a particularly stubborn program?

Send us the name of the program and its manufacturer and we'll add it to our Most Wanted List, a column (updated each issue) which helps to keep COMPUTIST readers informed of the programs for which softkeys are MOST needed. Send your requests to:

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

If you know how to deprotect unlock, or modify any of the programs below, let us know. You'll be helping your fellow COMPUTIST readers and earning MONEY at the same time. Send the information to us in article form on a DOS 3.3 diskette.

Apple Business Graphics	Apple Computer
Visiblend	Microlab
Gutenburg Jr. & Sr.	Micromation LTD
Prime Plotter	Primesoft Corp.
The Handlers	Silicon Valley Systems
The Apple's Core: Parts 1-3	The Professor
Fun Bunch	Unicorn
Willy Byte ...	Data Trek
Snoggle	Broderbund
ABM	Muse
Agent U.S.A.	Scholastic
Handicapping System	Sports Judge
Odin	Odesta
Mabel's Mansion	Datamost
Brain Bank	The Obsevatory
Crimson Crown	Penguin
Crypt of Media	Sir Tech
The Works	First Star Software
Cross Clues	Science Research
Peeping Tom	Microlab
Jigsaw	Microfun
Miner 2Ø49er II	Microfun
Create with Garfield	DLM
Print Master	Unision World
Bandits	Sirius Software
Bank Street Filer	Broderbund
Operation Frog	Scholastic Software
Work Force][	Core Concepts
Super Boulder Dash	Electronic Arts
Zorro	Datasoft
Goonies	Datasoft

readers' softkey & copy exchange

Larry Rando's softkey for...

Discovery!

Milliken Publishing Company
1100 Research Blvd.
POB 21579
St. Louis, MO 63132-0579

Requirements:
System Master (COPYA)
Any Sector Editor

Discovery (the scientific reasoning software) is an educational game, it is six sides, five of which contain two educational type games each. It has 10 problems in all plus a tutorial. The games consist of mind boggling, problem solving and others that will keep students busy for hours.

The Protection

Using COPYA did not work, so I disabled DOS's read error checking routine (B942:18) and successfully converted the disk to a readable format. I booted Disk 1 (the disks are self bootable) and the program read in a little bit of data on Track $00 but then started grinding. The grinding gave me an indication that the program was looking for something it did not find. I set up a boot trace. Tracing through the boot code I noticed that one of the epilogue bytes was off by one (DF AA EB opposed to the normal DE AA EB) this was on track 0, sector 2. I noted where it was so that it could be patched later since I had a hunch they changed some more, which they did. On track 0, sector 3 the same thing was done but this time two epilogue bytes were modified. No problem, I got up my sector editor and changed the modified epilogue bytes to what they should be DE instead of DF. No other occurance of this type of protection was around so I booted up and the program read in some more data but suddenly the drive rebooted. Hmm, a nibble count!

Indeed there was. I replaced the JMP $C600 (reboot routine) to a JMP $FF59 (jump to monitor) so that when I booted up the program it would break into the monitor and all the code would be intact. I examined the code especially around the $9B00 area. Since the JMP $C600 was not too far away, the nibble count had to be before this jump. A few bytes before the JMP $C600 was a JMP $9D84. I took a trivial guess thinking that this was the nibble count so I did a 9D84G, the drive turned on and the program worked fine. Great, I'm another step further, all that had to be done was find the nibble count and do something trivial like replace it with the JMP $9D84. The code around $9B00-FF (track 0, sector 0A) looked pretty strange so I followed each routine. At $9B08 was a JMP $AD99. Examining this routine I found tht it went to the $C600. I did a AD99G and the drive rebooted. I replaced it with the $9D84, did a 9B00G and the program loaded in and worked fine.

Step by Step

1) Boot normal DOS 3.3

2) Enter the monitor and Disable DOS's read error checking routine so that it will ignore the different epiloges.

CALL-151
B942:18

3) Use COPYA to copy disks 1-5 (Disk 6, the Manager disk, is unprotected).

4) Get out your sector editor and make the following changes to all the sides you just copied (they use the same exact protection).

Track	Sector	Byte	From	To
$00	$02	9E	DF	DE
$00	$03	35	DF	DE
$00	$03	91	DF	DE
$00	$0A	09	99	84
$00	$0A	10	AD	9D

5) Write it back out.

Dave Stanton's softkey for...

The M-ss-ng L-nks series

Sunburst Communications
39 Washington Avenue
Pleasantville, NY 10570

Requirements:
A way to Reset into the monitor, or
128K //e or //c with Xfer.boot/Restore
Original version of M-SS-NG L-NKS
Super IOB 1.5
Blank Disks

Computing educators soon learn to recognize the software publishers that consistently produce sound educational software. One of those companies is Sunburst Communications, and their "Missing Links" series proves the point.

Each disk in the series provides a variety of reading comprehension segments which are presented with selected letters omitted. The student must use the letters provided to fill in the blanks based on his knowledge of sentence structure and spelling principles. The skill level can be adjusted to be challenging for ages nine to ninety.

Sunburst deserves credit for its liberal backup policy, its willingness to discount lab packs, and its offering of networkable versions of its software.

Still, there are many legitimate reasons to create a softkeyed version of the "Missing Links" disks. After following this procedure, for example, you will be able to modify the reading selections to make them easier or more difficult. Furthermore, you will not be inconvenienced in the event that your originals suffer damage.

The Protection

Using a nibble editor to analyze the disk's raw nibbles reveals many changes to standard DOS formatting. The address prologues are D5 AA xx, where the value of the third byte is different on each track. The data prologues are similarly non-standard. Instead of the expected FF sync bytes, you'll find an unpredictable collection of values. Furthermore, these strange sync bytes cause confusion when you try to determine the data epilogues.

All things considered, capturing the disk's RWTS seems the most prudent approach to deprotection.

The Softkey

1) INIT a blank disk.

INIT LOGO

2a) If you have a copy card or other way to break into the montor, boot the original disk and allow it to continue loading files until it stops and waits for keyboard input. Then stop the program and move the RWTS to a safe area.

1900<B800.BFFFM

Now boot a 48K slave disk with no HELLO program and skip to step 6.

2b) If you have 128K, boot DOS 3.3 and load XFER.BOOT.

PR#6
BLOAD XFER.BOOT

3) Insert the original "Missing Links" disk, and boot it into your extended 80 column card.

PR#3
CALL 768

4) Allow the disk to continue loading files until the drive stops and waits for keyboard input. Hit Reset and go into the monitor to activate RESTORE.

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

5) Move the trapped RWTS from the 80 column card down to a safe location in the main 64K.

1900<B800.BFFF⌃Y

6) Save the protected RWTS on the same disk that holds Super IOB 1.5.

BSAVE RWTS.ML,A$1900,L$800

7) Load Super IOB 1.5 and insert the modified swap controller at the end of this article. RUN to make your copy from the original onto your already initialized backup disk.

Now that you have a completely deprotected version of "Missing Links", you may wish to add a faster operating system or change the literature text files to something of your own choosing. Doing so will require nothing more than a word processing program that accepts standard text files. Just be careful to maintain the text file format that the original uses. Imagine the possibilites!

controller

```
1000 REM MISSING LINKS CONTROLLER
1010 TK = 3 :LT = 35 :ST = 15 :LS = 15 :CD = WR :FAST
     = 1
1020 GOSUB 360 : GOSUB 490 : GOSUB 610
1030 GOSUB 360 : GOSUB 490 : GOSUB 610 : IF PEEK
     (TRK ) = LT THEN 1050
1040 TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO 1020
1050 HOME : PRINT "COPY^ DONE!" : END
10010 PRINT CHR$ (4 ); "BLOAD^ RWTS.ML,A$1900"
```

controller checksums

```
     1000  - $356B      1040  - $0D4F
     1010  - $2445      1050  - $E40C
     1020  - $C908      10010 - $8BE2
     1030  - $FB9C
```

Edward Hauff's softkey for...

Donald Duck's Playground

Sierra On-Line

Requirements:

Apple][
COPYA or fast copier
Sector editor
A blank disk
Donald Duck's Playground

Donald Duck's Playground, being a children's program, should be unprotected for multiple backups. No child is willing to wait for several weeks for a new disk to arrive after they have opened the disk jacket to find out where Donald Duck is hiding. Fortunately for us, On-Line has provided this disk in a near DOS 3.3 format with only a nibble count for protection. When the disk is copied it boots and everything runs fine until you turn the disk over. This is when the nibble count routine is invoked to check for the original. Stopping the routine at just the right moment is imperative as the program first checks to see if the disk has been turned over. If so, it calls the nibble routine and if that fails then the program will crash. After several tries with my trusty Replay II card I was able to find the nibble count in high memory hidden way back in the $FF00 page. Tricky, real tricky, but not quite tricky enough. By scanning this area, which by all rights and means should have been ROM, we will find the start of the nibble count at $FF07. Because the nibble count is only invoked after the program checks to be sure the disk has been turned over we must examine the code which does this check to see how it calls the nibble count. To do this we simply do not turn the disk over when the program asks. Press any key, and the check is made for the back, which of course fails. Next we try to catch this routine in action with our Replay card, where we will find a friendly JSR (Jump SubRoutine) to the nibble count. Armed with this information and a sector editor we search the disk for $20 07 FF which we will just NOP with three $EAs. Our effort is rewarded at turnover time with an operational and unprotected disk.

Here it is, short and sweet:

1) Copy the disk with COPYA or a fast copy program.

2) Search the disk for $20 07 FF. Mine was on track 2, sector 8, byte $2A.

3) Change the $20 07 FF to $EA EA EA. Write the sector back to the disk.

4) All done. Have fun.

Marc Batchelor's revisited softkey for...

Mastering The SAT

CBS Software
One Fawcett Place
Greenwich, CT 06836
$150.00

Requirements:

Apple][Plus, //e or clone
Whole disk copier that ignores errors
Four blank disks
Sector editor

Ths softkey for "Mastering the SAT" that was published in COMPUTIST No. 30 was unfortunately not very complete. First of all, the author failed to accomplish some things, while at the same time, he overlooked others. On page eight, the author states that "Tracks $03 and $04 have been written in a strange format." He also states that "The [Super IOB] controller copies tracks $05-$22 ... Use the same procedure on the other disks." Well, while it is true that on sides 1 and 2, tracks 3 and 4 have been formatted in some non-standard way, here is where the author made the mistake of assuming too much. He assumed that on sides 3 and 4, the same tracks were formatted the same way. **Wrong.** Only track 3 on sides 3 and 4 were formatted strangely. On track 4 of side 3, you will find 4 text files. They are READER 3, 4, 5, and 6. On track 4 of side 4 you will find the text files named V2SKILL5 and GR98. If you use the authors' softkey, you will lose a total of six text files.

I then realized that if I wanted to examine the contents of the files on my disks, I had to POKE things into memory, namely POKE 44033,21, LOAD the file, restore 44033 to a 17, and then examine the file. The softkey outlined by the author didn't take care of this inconvenience.

Two other things I found unsatisfactory were that when the Reset key is pressed from within the program it re-boots, and when I LIST the HELLO program, line zero contained some control characters which unless bypassed, will clear memory.

The softkey I will present here will solve all of the problems I have listed above and also make it readily available for installation on a hard disk.

The Softkey

My first problem was deciding what program to copy the originals with. I finally chose Locksmith fast copy to copy all four sides and ignore all errors. At this point, Mastering the SAT was in a COPYAable format. However, this still wasn't the format I wanted. I wanted hastle-free access to the programs.

I then went to track $15, sector $00 to examine its VTOC. This VTOC appeared perfectly normal, with the exception that it showed all tracks and sectors as being used. The VTOC even pointed to track $11, sector $0F as the first entry of the CATALOG which is the normal place for it. The normal place for the VTOC is track $11, sector $00. So I read in that sector to see what Mastering the SAT was storing where the VTOC should be. I was very happy to see that the entire sector was zeros. I therefore, read the VTOC from track $15, sector $00 and wrote it where it is supposed to go.

Putting a normal DOS on the disks then yielded a completely unprotected backup.

Futher examination revealed that this procedure worked for all four sides.

I then eliminated the last inconviences by LOADing the HELLO file from the unprotected copy and first typing "0" [RETURN] to eliminate the codes that clear memory and secondly removing the "POKE 1011,xxx : POKE 1012,xxx : POKE 214,255" from the program. This needed to be done to all of the disks.

Using a fast DOS produced a completely deprotected disk that boots about twice as fast as the original disk.

Step by Step

1) Initialize four disk sides (preferably with a fast DOS) with the boot filename of HELLO.

```
INIT HELLO
```

2) Use a disk copier that ignores errors to copy tracks $3 - $22 of all four sides of Mastering the SAT. If you are using Locksmith Fast Copy you need to use the following keystrokes to tell it to start at track $3.

```
0008⌃M 03⌃M
```

3) Use your sector editor to read track $15, sector $00 and write it to track $11, sector $00 of all four sides.

4) Remove the offending lines (lines 2-4) from the HELLO files of the four sides. In addition to the following commands, you should delete any commands from lines 2 through 4 which are similar to "POKE 1011,xxx : POKE 1012,xxx : POKE 214,255"

```
LOAD HELLO
0    (remove POKE sequences)
SAVE HELLO
```

That's all. Enjoy freshly deprotected diskettes. I must add that without the knowledge that there was no call for a nibble count (information supplied by Danny Pollak COMPUTIST No. 30), I might still be searching the disk for some code that tries to read tracks $3 and $4. Thanks for doing a lot of the leg work for the softkey, Danny.

Marc Batchelor's softkey for...

Copy II Plus 4.4C

Requirements:
A blank disk
Copy II Plus 4.4C
A disk searcher and sector editor
A way to break into the monitor

In an earlier issue of COMPUTIST, (Issue No. 11), Dr. Phillip Romine stated that Copy II Plus 4.4C will copy itself with parameters left at their default settings". Such was not the case with my copy of 4.4C. As a matter of fact, the parameters that were supplied with my version of CII+ 4.3 would not work on my 4.4C either. However, setting parameter 3A = 2 enabled me to make a copy. This worked well until my brother bought a hard disk from First Class Peripherals. He wanted to see if he could put CII+ 4.4C on his Sider hard disk. Well, needless to say, he was unsuccessful, leaving me to develop this softkey.

The procedure consists of capturing the utility part of the program, booting a slave disk and patching the utility program to restore itself when it is BRUNed. Then the bit copier is captured, the slave disk is rebooted, a small patch is made to the bit copier (to shut off the drive), and the program is saved.

In cookbook form:

1) Boot DOS 3.3 or your favorite fast DOS.

```
PR#6
```

2) INITialize a blank disk with an empty HELLO program.

```
FP
INIT HELLO
```

3) Boot your Copy II Plus 4.4C disk.

4) When the boot is completed, reset into the monitor by your favorite method.

5) Move the memory at $800-900 and $B000-BFFF to a safe place.

```
6000<800.900M
4C00<B000.BFFFM
```

6) Boot the slave disk you made at the start.

```
6⌃P
```

7) Enter the monitor and add a routine to move the memory back at run time.

```
CALL-151
800<6000.60FFM
82B:A9 00 85 00 85 02 A9 4C
833:85 01 A9 B0 85 03 A0 00
83B:B1 00 91 02 C8 D0 F9 E6
843:01 E6 03 A5 01 C9 5C D0
84B:EF A9 60 8D FF 02
```

8) Save the utility portion of the program onto the slave disk.

```
BSAVE CII+ 4.4C UTILITY,
A$82B,L$53FB
```

9) Now to get the bit copy portion of the program. Start by rebooting Copy II Plus 4.4C again.

```
6⌃P
```

10) Select "Bit Copy" from the main menu. When the drive stops again, break into the monitor.

11) Move the code at $800-900 to a safe place.

```
6000<800.900M
```

12) Boot the slave disk.

```
6⌃P
```

13) Enter the monitor and make a patch to the bit copier.

```
CALL-151
800<6000.60FFM
808:A9 60 8D FF 02 4C 00 09
```

14) Save the bit copier to the slave disk.

```
BSAVE CII+ 4.4C BIT COPY,
A$808,L$2AFB
```

15) Boot your sector editor/disk searcher. We will make a patch to the utility program so that selecting the bit copier will put you in the monitor instead.

16) Scan the slave disk for the sequence $BD 89 C0 A9 00.

17) Count 3 bytes backward from the byte $BD and type:

```
2C E8 C0 4C 59 FF
```

18) Write the sector back to the disk.

19) You may wish to create a Hello program on the disk to select which part of Copy II Plus you want to use.

Congratulations. You are now the proud owner of a version of Copy II Plus 4.4C that can be run from a hard disk.

Mike Landen's softkey for...

Master of the Lamps

Activision, Inc.
2350 Bayshore Frontage Rd.
Mountain View, CA 94043

Requirements:
A fast copier
A sector editor

The Procedure:

1) Copy Master of the Lamps to a blank disk using and fast copier.

2) Use your sector editor to change track $00, sector $0B, bytes $25-29, to A9 FF 4C 6A A1.

That's it...

readers' softkey & copy exchange

Blaine Gardner's softkey for...

One on One

Electronic Arts
2755 Campus Drive
San Mateo, CA 94403

Requirements:
Apple][
Super IOB 1.5
A blank disk

A softkey for One on One was printed in COMPUTIST No. 13, but it didn't work on my version, which was purchased last month.

After some tinkering I discovered that its protection is similar to Archon II, which was covered in COMPUTIST No. 24. The nibble count is on tracks 5 & 6, and the data markers are the same (D5 BB CF), but the sector edits for the nibble count routine didn't work.

Instead of trying to do all of the work myself, I tried a trick which I recently discovered: the Copy II Plus parm file! For example, I copied Financial Cookbook with the Auto Bit Copy option, and was quite surprised that my copy was now deprotected!! Examining the parm file for Financial Cookbook showed that it was a normal copy combined with a sector edit that disabled the nibble count.

It wasn't that easy for One on One because of the altered data markers. But, combining the sector edits from Copy II Plus (6.0) with the controller for Archon II provided a simple and painless method of deprotecting One on One.

I have looked through several of Copy II Plus' parm files, and if the disk is protected with a nibble count Copy II Plus stands a pretty good chance of cracking it. You may have to normalize the DOS, but that is usually the easy part. If you're having a tough time tracking down a nibble count routine, give Copy II Plus a try. It can save an awful lot of work!

To softkey the newest version of One on One, simply install this controller in Super IOB 1.5 and run it as usual.

controller

```
1000 REM ONE ON ONE (NEW VERSION)
1010 TK = 0 :ST = 0 :LT = 35 :CD = WR
1020 T1 = TK : GOSUB 490 : IF TK > 3 THEN RESTORE
     : GOSUB 210
1030 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     DOS THEN 1030
1035 IF TK = 2 THEN GOSUB 210
1040 IF BF THEN 1060
1050 ST = 0 :TK = TK + 1 + (TK = 4) * 2 : IF TK <
     LT THEN 1030
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 + (TK = 4) * 2 : IF BF =
     0 AND TK < LT THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "DONE^ WITH^ COPY" : END
5000 DATA 213 ,187 ,207
5010 DATA 13^ CHANGES
5020 DATA 2 ,3 ,71 ,170
5030 DATA 2 ,3 ,81 ,173
5040 DATA 1 ,6 ,8 ,98
5050 DATA 12 ,4 ,0 ,160
5060 DATA 12 ,4 ,1 ,3
5070 DATA 12 ,4 ,2 ,162
5080 DATA 12 ,4 ,3 ,255
5090 DATA 12 ,4 ,4 ,169
5100 DATA 12 ,4 ,5 ,9
5110 DATA 12 ,4 ,6 ,24
5120 DATA 12 ,4 ,7 ,96
5130 DATA 12 ,4 ,8 ,4
5140 DATA 12 ,4 ,9 ,30
```

controller checksums

```
1000  - $356B     5020  - $F9DD
1010  - $3266     5030  - $D9AF
1020  - $2E67     5040  - $528D
1030  - $3866     5050  - $CB13
1035  - $7B0B     5060  - $C770
1040  - $2B80     5070  - $BCFC
1050  - $7275     5080  - $F823
1060  - $0803     5090  - $9CD3
1070  - $0006     5100  - $0580
1080  - $8D0D     5110  - $EC96
1090  - $4CD7     5120  - $923A
1100  - $00A8     5130  - $1506
5000  - $8E56     5140  - $5143
5010  - $FCA3
```

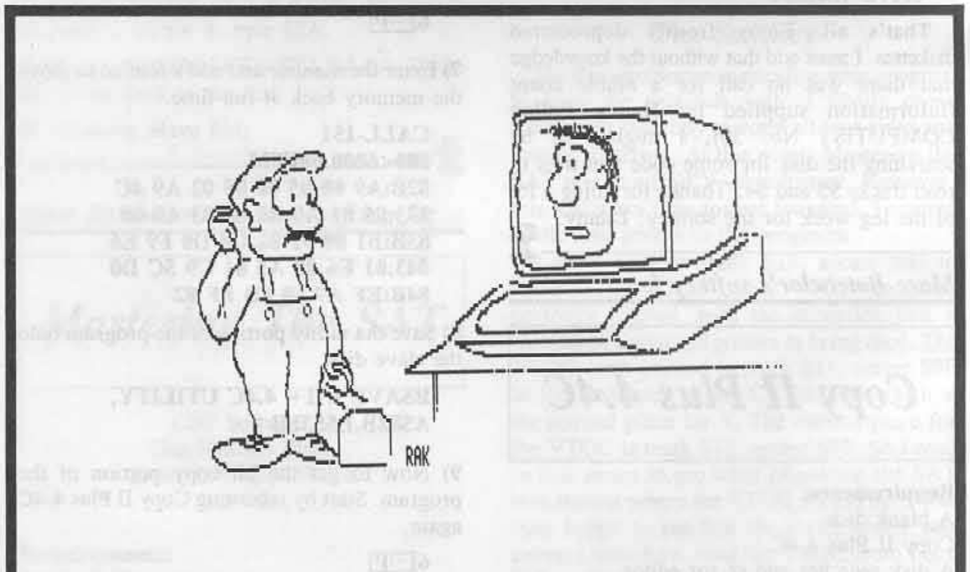
Will Knight's softkey for...

Bridge Baron

Great Game Products
P.O. Box 76
Cabin John, MD 20818-0076

Requirements:
Bridge Baron
64K RAM
Super IOB 1.5
A blank disk

Bridge Baron is, hands down, the best bridge program available for the Apple. Bidding supports the most common conventions and play of the hand is not too bad either (don't expect to see a Vienna coup, though!). I paid over $50 for the program; software this expensive MUST be backed up. Bit copiers wouldn't touch it - not even the highly touted (and priced) Echo Plus. However, the appearance of an Applesoft prompt during the boot was a hopeful sign. Normally I just slap in Super IOB with a swap controller and let 'er rip, but that also failed. Upon examining the



"The directions said to enter the monitor"

RWTS I found it so modified that Super IOB appeared doomed. $B800-$B956 was filled with PLP's and BRK's and other garbage. The RWTS appeared to start at $B957, but swapping it with that address start in lieu of $B800 was to no avail.

After a while, I noticed that Bridge Baron's protection looked a lot like Moebius, so I tried the softkey for that program which appeared in COMPUTIST No. 32. When applied to Bridge Baron it worked fine; I didn't even have to make any sector edits since Bridge Baron never writes to the disk. The take-home lesson here is that not only can you deprotect different programs from the same company with the same method, since copy protection is often marketed to several publishers, you can also sometimes deprotect programs from different companies with one method.

Step by Step

1) Initialize a disk with a DOS of your choice:

INIT BRIDGE BARON 10/15/85

2) Run Super IOB 1.5 with the controller for ULTIMA IV (COMPUTIST No. 28, pg. 10) using the disk prepared in step 1 as the destination. Note: the POKE 47829, 13 in line 1060 of the controller should be POKE 47829, 213 as pointed out in Jeff Rivett's Moebius softkey.

Finally, a partner that won't pass your jumpshifts! Enjoy.

Phil Goetz' softkey for...

A. E.

Broderbund Software
17 Paul Drive
San Rafael, CA 94903

Requirements:
A way to reset into the monitor
A 48K slave disk with no HELLO
An INITialized, double sided blank disk
COPYA

The AE disk has two sides: the boot side and the scenario side. Only the boot side is protected. To unlock AE, we will save the boot program as a file.

Install an old F8 ROM or some other means of resetting into the monitor. Boot AE, press space, and wait for it to tell you TURN THE DISKETTE OVER AND PRESS SPACE. Then stop the program. Enter the following from the monitor:

```
20F7<7F7.8FFM
2200<0.FFM
2300<8B00.A7FFM

2000:A0 00 A9 D4 99 00 A8 C8
2008:D0 FA EE 06 20 AD 06 20
2010:C9 C0 D0 EE A9 F7 B9 00
2108:20 99 00 07 C8 D0 F7 EE

2020:18 20 EE 1B 20 AD 1B 20
2028:C9 09 D0 EA A0 00 B9 00
2030:22 99 00 00 C8 D0 F7 A0
2038:00 B9 00 23 99 00 8B C8

2040:D0 F7 EE 3B 20 EE 3E 20
2048:AD 3E 20 C9 A8 D0 EA A6
2050:00 AD 50 C0 AD 57 C0 AD
2058:54 C0 AD 52 C0 20 93 FE
2060:20 89 FE 4C 00 90

2080:A9 00 8D F4 B7 8D EB B7
2088:8D EC B7 8D F5 B7 8D F3
2090:B7 20 E3 03 20 B5 B7 4C
2098:00 20
```

This disassembles to

```
2000- LDY   #$00
2002- LDA   #$D4
2004- STA   $A800,Y  Fill memory with $D4
2007- INY
2008- BNE   $2004
200A- INC   $2006
200D- LDA   $2006
2010- CMP   #$C0
2012- BNE   $2002
2014- LDY   #$F7
2016- LDA   $2000,Y
2019- STA   $0700,Y  7F7<20F7.21FFM
201C- INY
201D- BNE   $2016
201F- INC   $2018
2022- INC   $201B
2025- LDA   $201B
2028- CMP   #$09
202A- BNE   $2016
202C- LDY   #$00
202E- LDA   $2200,Y
2031- STA   $0000,Y  0<2200.22FFM
2034- INY
2035- BNE   $202E
2037- LDY   #$00
2039- LDA   $2300,Y
203C- STA   $8B00,Y  Move 8B00.A7FF back
203F- INY
2040- BNE   $2039
2042- INC   $203B
2045- INC   $203E
2048- LDA   $203E
204B- CMP   #$A8
204D- BNE   $2039
204F- LDX   $00
2051- LDA   $C050
2054- LDA   $C057    HGR
2057- LDA   $C054
205A- LDA   $C052
205D- JSR   $FE93    Disconnect DOS
2060- JSR   $FE89     "       "
2063- JMP   $9000    Start AE

2080- LDA   #$00
2082- STA   $B7F4    Seek
2085- STA   $B7EB    Volume 0
2088- STA   $B7EC    Track 0
208B- STA   $B7F5    Clear error flag
208E- STA   $B7F3    Byte count
2091- JSR   $03E3    Set up for RWTS
2094- JSR   $B7B5    RWTS
2097- JMP   $2000    Memory moves (above)
```

Note that the first thing this boot program does is move the drive head to track 0. This is the cleverest part of the protection on AE: loading the protected boot program leaves the head on track 0, and when you flip to the scenario disk, the head must still be on track 0 to run. Now boot a disk with no hello program or a very short one, and enter the following:

```
CALL-151
8FD:4C 80 20
A964:FF
BSAVE AE,A$8FD,L$8210
```

(A964:FF lets you save long files)

Now, when you BRUN AE, instead of asking you to "TURN THE DISKETTE OVER AND PRESS SPACE" the screen will switch to hi-res. Turn the disk over and press the space bar anyway.

Larry Rando's softkey for...

Great American Cross-Country Road Race

Activision, Inc.
2350 Bayshore Frontage Rd.
Mountain View, CA 94043

Requirements:
COPYA
A sector editor

The Protection

After copying the disk with a fast copier, I booted up and got as far as the Activision logo (The loading process is very long and takes a while). After displaying the logo the drive started recalibrating in an endless loop. It seemed it was looking for something it could not find. After some time boot tracing I came across a funny disk-read routine(s) which I found caused the disk drive recalibration. Besides the abnormal disk read routine (which was almost half a sector long), the program did a checksum of the bytes. If it did not add up to $FC, it would of course crash. To correct it I had to NOP out all of $C2-$FF which had nothing to do with the boot process, although it was covered up nicely to loop legitimately. Finally, I changed the byte that contained the final checksum of the routine (from FC to FF).

The Procedure

1) Copy the disk with any fast copier.

2) Start up your favorite sector editor and make the following changes:

```
Track Ø, Sector $A:
  Change bytes $C2-$FF all to EAs.

Track Ø, Sector $B:
      Byte     From     To
      -------  -------  ----
      $ØØ       38      EA
      $Ø1       2A      EA
      $Ø2       25      A9
      $Ø3       FC      FF
```

Dan Agnew's softkey for...

Computer Preparation for the SAT

Harcourt, Jovanivich, Brace

Requirements:
COPYA
A sector editor

Computer Preparation for the SAT is a very good preparation program. There's only one problem: it's copy-protected, making the loading time very slow.

The procedure to make a COPYAable version is as follows:

1) Defeat DOS' error checking and return to BASIC.

```
CALL -151
B942:18
3DØG
```

2) Now copy each disk with COPYA.

3) Make the following change to each of the COPYAed disks with a sector editor:

```
  Track   Sector   Byte   Change To
  ---------------------------------
   $11     $Ø       $2      $ØF
```

4) Add some fast DOS or DOS 3.3 to the disks and enjoy.

Jim Hart's softkey for...

Castle Wolfenstein

Muse Software
347 N. Charles St.
Baltimore, MD 212Ø1

Requirements:
48K or more
One or more disk drives
Super IOB 1.5
A blank disk
A DOS 3.3 compatible fast DOS (optional)
Original Castle Wolfenstein disk

Castle Wolfenstein is one of the classics in the computer gaming world. However, on the Apple series, it is copy protected, and this plays havoc for us "Wolfenstein junkies". It takes about one month for the disk to wear out (for me at least) and sending off for backups from Muse takes both time and money. I decided to try to deprotect the disk and possibly add a quickie DOS to speed things up a bit. Guess what? I succeeded!

The first thing I did was use Copy II Plus to see which tracks had valid data on them. Hmmmm... it seems all 35 tracks are being used. Number two on the list is to look at a raw nibble dump and check to see if there are any strange prologues or epilogues in either the address or data fields. Aha! The addresss prologue is changed to D5 DA 96 and the data prologue is changed to D5 DA AD. Additionally, the checksums seemed to be incorrect in several data sectors also. At this point it appeared that I could incorporate this information into a Super IOB controller to deprotect the disk. If you want to see the preliminary results, try the Super IOB controller given here with the POKEs to location 47786 and use it on the Castle Wolfenstein disk. The copy seems to be A-OK but just for kicks try to CATALOG it. An I/O error, you say? I was stumped at this point. However, looking through my back issues of COMPUTIST helped yield the solution. In COMPUTIST No. 21, there is a softkey for Advanced Blackjack and in it Jim Mitchell describes how a change in the byte translate tables can foul up a copy. I checked out the areas in DOS that he mentions and sure enough, it is altered on the Wolfenstein disk. Incorporating this one change (to location 47786) into the controller gives a perfectly working copy.

Step by Step

1) INITialize a blank disk, preferably with a fast DOS, using the name HELLO.

2) Use the controller below with Super IOB 1.5 to copy the Wolfenstein disk to the newly initialized disk.

3) After your disk has been deprotected, type in this little program and save it to the copied disk under the name HELLO.

```
FP
1Ø HOME : D$ = CHR$(4)
2Ø PRINT D$; "BRUN @INIT"
SAVE HELLO
```

4) You now have a COPYAable disk. Hide the original and enjoy!

controller

```
1ØØØ REM CASTLE WOLFENSTEIN CONTROLLER
1Ø1Ø TK = 3 : LT = 35 : ST = 15 : LS = 15 : CD = WR : FAST
     = 1
1Ø2Ø RESTORE : GOSUB 19Ø : GOSUB 21Ø : POKE 47426
     , 24 : POKE 47786 , 35 : GOSUB 49Ø : GOSUB 61Ø
1Ø3Ø GOSUB 23Ø : POKE 47426 , 56 : POKE 47786 , 17Ø
     : GOSUB 49Ø : GOSUB 61Ø : IF PEEK (TRK ) = LT
     THEN 1Ø5Ø
1Ø4Ø TK = PEEK (TRK ) : ST = PEEK (SCT ) : GOTO 1Ø2Ø
1Ø5Ø HOME : PRINT "COPY^ DONE. " : END
5ØØØ DATA 213 , 218 , 15Ø , 213 , 218 , 173
```

controller checksums

```
    1ØØØ  - $356B       1Ø4Ø  - $4D6A
    1Ø1Ø  - $2445       1Ø5Ø  - $363B
    1Ø2Ø  - $F6C6       5ØØØ  - $1EØC
    1Ø3Ø  - $7FBB
```

Mike Landen's softkey for...

The Luscher Profile

Mindscape Inc.
3444 Dundee Rd.
Northbrook, IL 6ØØ62

Requirements:
COPYA or equivalent
A sector editor

1) Copy both sides of Luscher Profile to a blank disk using any fast copier.

2) Use your sector editor to change track $12, sector $6, bytes $24-$25, from A9 44, to 18 6Ø. That's it...

readers' softkey & copy exchange

Marc Batchelor's softkey for...

Skyfox

Electronic Arts
39Ø Swift Avenue
South San Francisco, CA 9408Ø

Requirements:
Skyfox
A blank disk
Super IOB 1.5
Sector Eitor

In COMPUTIST No. 17 (pg. 7), Marshall Strouse wrote a softkey for Skyfox from Electronic Arts. Unfortunately, the softkey was lacking in a couple of aspects. When I used the softkey described in that article, the "trapdoor" to Alpha Invaders was lost. I don't own a Mockingboard, but I also understand that quite a bit of sound capability was lost. I have a different method to remove the copy protection that at least leaves Alpha Invaders alone. The method uses the Archon controller from COMPUTIST No. 21.

1) Install the controller below into Super IOB 1.5 and run it on Skyfox.

2) Get out your sector editor and make the following changes to the copy.

Track	Sector	Byte	Change To
$1	$6	$3Ø	$Ø4
$2	$Ø	$AA	$6Ø
$2	$3	$47	$AA
$2	$3	$51	$AD
$13	$8	$69	$A9 4D 18 6Ø
$13	$8	$74	$BB C5 A7 97

There you have it, Skyfox in a bag. Remember to write the sectors back to the disk.

controller

```
1000 REM SKYFOX / ELEC ARTS
1010 TK = Ø :ST = Ø :LT = 35 :CD = WR
1020 T1 = TK : GOSUB 490 : IF TK > 3 THEN RESTORE
     : GOSUB 210
1030 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     DOS THEN 1030
1035 IF TK = 2 THEN GOSUB 210
1040 IF BF THEN 1060
1050 ST = Ø :TK = TK + 1 + (TK = 4 ) * 2 : IF TK <
     LT THEN 1030
1060 GOSUB 310 : GOSUB 230 : GOSUB 490 :TK = T1
     :ST = Ø
1070 GOSUB 430 : GOSUB 100 :ST = ST + 1 : IF ST <
     DOS THEN 1070
1080 ST = Ø :TK = TK + 1 + (TK = 4 ) * 2 : IF BF =
     Ø AND TK < LT THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT : PRINT "DONE^ WITH^ COPY" :
     END
5000 DATA 213 ,187 ,207
5010 DATA 2^ CHANGES ,2 ,3 ,71 ,170 ,2 ,3 ,81 ,173
```

controller checksums

```
1000  - $356B      1060  - $0803
1010  - $3266      1070  - $0006
1020  - $2E67      1080  - $8D0D
1030  - $3866      1090  - $4CD7
1035  - $7B0B      1100  - $3107
1040  - $2B80      5000  - $E127
1050  - $7275      5010  - $161B
```

Greg Poulos and Jim Hart's softkey for...

Silent Service

Requirements:
Copy program that ignores errors
Disk Searcher
Sector Editor

Silent Service is a submarine simulation game. It has proved to be very easy to deprotect. Presented here are two possible methods:

Procedure #1

1) Copy the original disk with Locksmith Fast Copy (or some other whole disk copier that will ignore errors).

2) Read in track $Ø4, sector $Ø9 with a sector editor.

3) Somewhere around bytes $DØ-$EF, you should see some text that says something to the effect of "HARDWARE FAILURE". If you don't see this text, search the disk for it.

4) Enter the following codes with the first byte covering up the "H" in "hardware".

ØØ 84 C2 CC CF C1 C4 C7
8D 84 C2 D2 D5 CE C1 8D

Procedure #2

1) Copy the original disk with Locksmith Fast Copy (or some other whole disk copier that will ignore errors).

2) Using a disk searcher, scan the disk for the sequence:

9A DB 8D Ø1 Ø2 4C AA Ø2 A9 Ø2

You should find it somewhere in the file "\\". On my disk this is was on track $Ø5, sector $Ø9, byte $AA.

3) Change the last byte in the sequence to a $DB.

4) That's it! Another victory in the deprotection wars.

readers' softkey & copy exchange

Jeff Wicker's hints for...

Carmen Sandiego

What's that I hear? You say you're tired of looking up places, currency, and landmarks in Carmen Sandiego? Well, I've got just the thing for you. A little cheat table. On the left side of each list is the name, landmark, etc., and on the right is the location to go to.

Currency

Francs	Paris
Drachmas	Athens
Rubles	Moscow
Lira	Istanbul or Rome
Pounds	London
Pesos	Mexico City

Landmarks

The Amazon	Rio
The Ganges	New Delhi
The Danube	Budapest
The Tigris	Bagdhad
Eiffel Tower	Paris
Mozambique Channel	Moroni
Mt. Popocatepeti	Mexico City
Mt. Olympus	Athens
Mt. Blanc	Paris
Loch Ness	London
Sikh Temples	New Delhi
Shinto Shrines	Tokyo
United Nations	New York
Louve	Paris
Sahara	Bamako
Big Ben	London
Skyscrapers	New York
Subways	New York
Black Sea	Istanbul

Culture

Aztec	Mexico
Magayar	Budapest
Georgian	Moscow
Etruscan	Rome
Byzantine	Istanbul
Sparta	Athens
Bedouins	Cairo
Serendip	Colombo
Sumeria	Baghdad
Hittite	Istanbul
Saxon	London

Products

Rice	Colombo
Marble	Rome
Wine	Paris
Tweed	London
Whiskey	London
Textiles	Paris
Fashion	Paris
Stamp Buyer	San Mario
Diamonds	Rio
Gorilla	Kigali

Helpful Hints

A robber never (at least in any of my games) goes back to a location where he/she has been before during the course of that game. When you first start to investigate go to all three places in your investigating option until you have gathered three or four pieces of information to plug into the interpool. It is very important to get a warrant as soon as possible.

After this is done, the cheat sheet becomes most handy because you may then zip from place to place until you find the suspect. After traveling to a location, if your first investigation results in a rejection travel back to the place where you just left and try going somewhere else. Good luck!

Danny Pollak's softkey for...

Echo Plus

Agranat Systems
1Ø Winthrop Circle
Weston, MA Ø2193
$59.95

Requirements:
Apple][Plus or equivalent
Echo Plus
A way to enter the monitor (see article)
INITialized disk with no HELLO program

Echo Plus is one of the newest copy programs to hit the market. There are no parameters to change, although it helps to know which tracks to copy. The automatic disk copier is a nice feature. It can be used to get a good idea of which tracks to copy and sometimes will even produce a good copy . The program comes with a 42 page manual which covers the options offered by the program and also talks about different copy protection schemes and offers some tips.

The Protection

Echo Plus is rather heavily protected. My first attempt at deprotecting the program was to see if it could be copied using Super IOB. An examination of the disk showed that it was written in a format which was far from normal, which rules out Super IOB. The program appears to lie between Tracks $ØØ and $Ø4 and it looks like it is stored on half, as well as, whole tracks.

The next thing I tried was to boot code trace the disk. All was well until the code started loading onto the zero page, the stack, the keyboard buffer and text page 1. At this point I decided that there must be a better and easier way to do this.

The third technique I tried was to boot the program into auxiliary memory. This took me nowhere. The program only partially loaded before crashing into the monitor.

I began to get desperate. I didn't own an NMI card and I didn't have any other way of resetting into the monitor... or did I? My brother has a II+ with a 16K RAM card. I remembered reading about how it was possible to reset into the monitor by moving the RAM card to slot 1 and setting it up to exit to the monitor. To make a long story a little shorter, I tried it and it worked.

The Procedure

Although this procedure uses a RAM card in slot 1 to enter the monitor, any other method is fine as long as you have the ability to save lower memory (pages Ø-7).

1) The first thing that you must do is to move your RAM card to slot 1. After moving the RAM card, insert Echo Plus in drive one and enter the following commands:

```
CALL-151
CØ91 CØ91
F8ØØ<F8ØØ.FFFFM
FFFC:ØØ BF
CØ9Ø
BFØØ:A2 ØØ BD ØØ ØØ 9D ØØ 1Ø
BFØ8:E8 DØ F7 EE Ø4 BF EE Ø7
BF1Ø:BF AD Ø7 BF C9 18 DØ EA
BF18:2C 91 CØ 2C 91 CØ 4C 59 FF
6⌃P
```

2) After the program has finished loading, hit Reset and you should end up in the monitor. Now we will compact the code for the purpose of creating the smallest file possible. Page 8 must be moved to upper memory so that it won't be wiped out when DOS is booted.

```
18ØØ<8ØØØ.99FFM
3258<9A8Ø.9ACBM
6ØØØ<8ØØ.8FFM
```

3) Boot the initialized disk. Enter some necessary code and save the program as a single binary file.

```
6⌃P
CALL-151
8ØØ<6ØØØ.6ØFFM
7FD:4C ØØ 32

32ØØ:A2 ØØ BD ØØ 1Ø 9D ØØ ØØ
32Ø8:E8 DØ F7 EE Ø4 32 EE Ø7
321Ø:32 AD Ø7 32 C9 Ø8 DØ EA
```

```
3218:BD ØØ 18 9D ØØ 8Ø E8 DØ
3220:F7 EE 1A 32 EE 1D 32 AD
3228:1D 32 C9 9A DØ EA BD 58
3230:32 9D 8Ø 9A E8 EØ 4F DØ
3238:F5 A9 AØ 8D ØØ 48

323E:2C 83 CØ 2C 83 CØ A9 ØØ
3246:8D ØØ 9A 8D 4Ø 9A 8D 1B
324E:FD 8D FØ FD A2 FF 9A 4C
3256:ØØ 8Ø
```

BSAVE ECHO PLUS,A$7FD,L$2AA7

That's it. Now that you have a backup of Echo Plus, you can use it to make backups of all of your other software.

Edward Hauff's softkey for...

Swashbuckler

Datamost

Requirements:
Apple][
A way to break into the monitor
Super IOB with swap controller
Swashbuckler disk
Blank disk

When I first played Swashbuckler I could not understand why Datamost would want to copy protect it. After several short attempts at playing it, I was convinced that it should be in standard format so when you got sick of this game you could have more fun erasing the disk. The protection on this is altered address and data marks which can be circumvented using the swap controller with Super IOB.

To make the inconvenience as short as possible:

1) INIT a slave disk with an empty Hello and delete same.

2) Boot the Swashbuckler disk and hold CTRL C while the boot is in progress.

3) When the Applesoft prompt appears break into the monitor with your favorite method.

4) Move the RWTS down to $1900.

1900<B800.BFFFM

5) Boot the slave disk you INITialized in step one.

6) Save the Swashbuckler RWTS to your Super IOB disk.

BSAVE SWASHBUCKLER RWTS ,A$1900,L$800

7) Run Super IOB and copy Swashbuckler to the disk you INITialized in step one.

8) All done, try to have fun.

Danny Pollak's softkey for...

Randamn

Magnum Software

Requirements:
Apple][Plus or equivalent
Randamn
Super IOB 1.5
A blank disk

The program Randamn from Magnum Software is a game in which you, a candidate for the replacement of the demi-god Randamn, must traverse seven different worlds, each with seven stages. You must complete all seven stages (28 randomly chosen opponents each) before you can advance to the next world. This means that you must defeat 196 randomly chosen opponents in order to win the game!

The Protection

An examination of the Randamn disk shows that the address and data epilogue marks have been changed from DE AA to DF FE on tracks Ø-$2Ø. The volume number of each sector has also been changed so that the address checksum will equal zero. The program checks to see that it is zero. Track $21 is used for the purpose of a nibble count and track $22 is unused. There is also a check during the boot to make sure that the sync bytes written before the address marks are not $FF or $FE.

After extensive searching, I was able to find the routines that verify that the address checksum is zero. A typical routine would look something like this:

```
LDA $2E    Load the current track #
EOR $2D    Exclusive-or with sector #
CMP $2F    Compare volume # to accumulator
BNE ...    Branch on result not equal
```

Using this, or a similar manner, it was possible for the protectors to verify that the address checksum was zero. Once the routines were found, all that had to be done was to make the check branch to the correct location even if the address checksum was not equal to zero.

The Softkey

1) Install the controller found at the end of this article into Super IOB 1.5.

2) RUN Super IOB and follow the prompts. Initialize the disk with a volume number of zero. The disk will be copied and the following sector edits will be made to prevent the checks from taking place:

Track	Sector	Byte	From	To
ØØ	Ø3	42	38	18
Ø3	ØØ	4D	C6	6Ø
Ø6	Ø9	26	F8	EA
Ø6	ØE	8D	96	ØØ
Ø7	Ø3	1A	F8	EA
Ø7	Ø7	7C	A9	EA
Ø7	Ø7	7D	ØC	EA
Ø7	Ø7	7E	85	EA
Ø7	Ø7	7F	67	EA
Ø8	Ø7	54	2Ø	4C
Ø8	Ø7	55	4D	BD
Ø8	Ø7	56	7D	64
Ø9	Ø8	D2	25	ØØ

controller

```
1000 REM RANDAMN CONTROLLER
1010 TK = Ø : LT = 33 : ST = 15 : LS = 15 : CD = WR : FAST
     = 1
1020 RESTORE : GOSUB 170 : GOSUB 490 : GOSUB 610
1025 T1 = TK : TK = PEEK (TRK ) : GOSUB 310 : TK = T1
1030 GOSUB 230 : GOSUB 490 : GOSUB 610 : IF PEEK
     (TRK ) < LT THEN 1050
1040 TK = PEEK (TRK ) : ST = PEEK (SCT ) : GOTO 1020
1050 HOME : PRINT "COPYDONE" : END
5000 DATA 223 ,254 ,223 ,254
5010 DATA 13^ CHANGES
5020 DATA Ø ,3 ,66 ,24
5030 DATA 3 ,Ø ,77 ,96
5040 DATA 6 ,9 ,38 ,234
5050 DATA 6 ,14 ,141 ,Ø
5060 DATA 7 ,3 ,26 ,234
5070 DATA 7 ,7 ,124 ,234
5080 DATA 7 ,7 ,125 ,234
5090 DATA 7 ,7 ,126 ,234
5100 DATA 7 ,7 ,127 ,234
5110 DATA 8 ,7 ,84 ,76
5120 DATA 8 ,7 ,85 ,189
5130 DATA 8 ,7 ,86 ,100
5140 DATA 9 ,8 ,210 ,Ø
```

controller checksums

```
1000  - $356B     5040  - $B279
1010  - $1A22     5050  - $F2F8
1020  - $7CA8     5060  - $72DD
1025  - $61B9     5070  - $A6C7
1030  - $D471     5080  - $58Ø3
1040  - $FC56     5090  - $Ø4ØC
1050  - $75FA     5100  - $5481
5000  - $6550     5110  - $41E6
5010  - $34D4     5120  - $6BB7
5020  - $919E     5130  - $ØABD
5030  - $3739     5140  - $7AEØ
```

Turn Drive 1 into Drive 2 with the...

Electronic Disk

by William Wingfield Jr.

Requirements:
Disk II controller card
Low-wattage soldering iron
Small hook-up wire
Miniature DPDT switch
14 pin IC socket or 74LS132 IC (optional)

Note: The procedure described below requires modification of the disk controller card and may (will) void any warranty. COMPUTIST will not be held responsible for any damages incurred while following this procedure.

If you have two disk drives there have probably been times when you wished there was a way to do a boot from drive 2. This article will show you how to make a simple hardware modification to your drive controller card that will allow you to do this at a flip of the switch. While this article is written using the Apple Disk II Interface Card for illustration, the techniques used apply to almost any floppy drive controller card.

Theory

A brief description of how the card accesses the drives will help in understanding how this modification works. Look at the schematic of the controller card on page 145 in the Apple DOS 3.3 manual and examine your card. All information to and from the drives is carried through the 2Ø pin connecting cables. A look at the schematic or card will show you that the pins for the cable connectors are all parallel except for the ones on pin 14. These are the ones that enable the selected drive. If we could reverse these two connections the controller would actually access drive 2 when it meant to access drive 1. This allows a boot from drive 2 and other uses that will be discussed later.

Both pin 14's of the cable connectors come from the 74LS132 chip at location A2 on the card; drive 1 from pin 18 of the chip and drive 2 from pins 6 and 9. We will make the modification here at the chip.

Procedure

Start by deciding where you would like the switch located. It may be allowed to hang loose out the back of the computer but I recommend mounting it on the case. Cut four pieces of hook-up wire long enough to reach from the card to the location that you have selected for the switch. Solder these to the DPDT switch along with two short pieces to make a reversing switch as shown in Fig. 1. (Note: You DON'T want a Center-Off type of switch.)

Next remove the controller card, disconnect the drive cables and remove the chip (the 74LS132 at location A2). Prepare it by gently bending pins 6, 8 and 9 straight out from the chip. Pin 6 must be bent up because a foil connection under the socket connects it to pin 9.

Remove 1/8 inch insulation from one of the common pins of the switch and solder it to pin 8 of the IC. Remove 5/8 inch insulation from the other common pin of the switch and solder it to pins 6 and 9 of the IC, bridging across the top of the chip with the wire. Make these connections quickly to avoid possible heat damage to the chip. "Tinning" the wires will allow you to make the connections quickly and without overheating the IC.

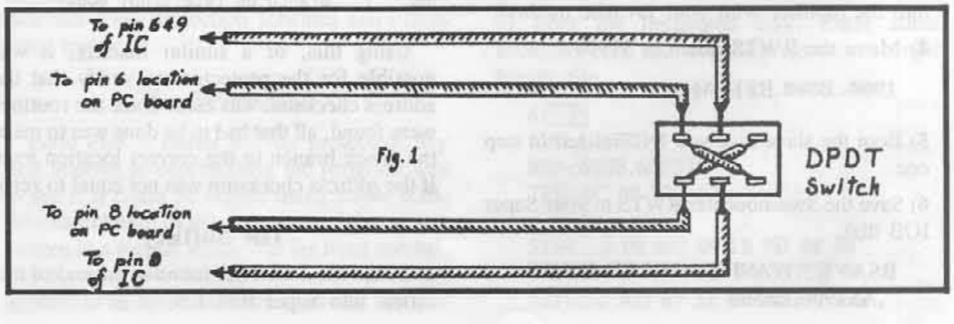


Fig. 1

Remove 1/8 inch insulation from the remaining two leads and solder these to the back of the controller card at pin locations 8 and 9; refer to Fig. 2 to identify these. If these conections are a little closer than you feel comfortable with you can use locations 8 and 6 because of the connection between 6 and 9 under the socket on the board.

Now place the chip back in the socket. Fig. 3 shows the chip in position with the pins bent out. The wires to the back of the card are omitted for clarity. Thoroughly inspect your work for proper wiring, solder bridges, etc. Once satisfied that everything is in order you may proceed to test the modified card.

Simply replace the modified card, reconnect your drive units and power up. One of the drives should come on as usual. Turn the power off, flip the drive reversing switch and power up again. This time the other drive should come on. Note which position of the switch gives the normal start up of drive 1 and which gives the new option of starting up with drive 2. Mark or position the switch so that you know which is which and you are done.

Alternatives

If you don't wan't to solder to the card itself you may simply insert the wires that solder to the card into their respective positions in the IC socket. Extra care must be taken to make sure that the leads make good contact in the socket and that they don't contact the pins that were bent up on the chip. A spare 74LS132 chip

Drive Swapper

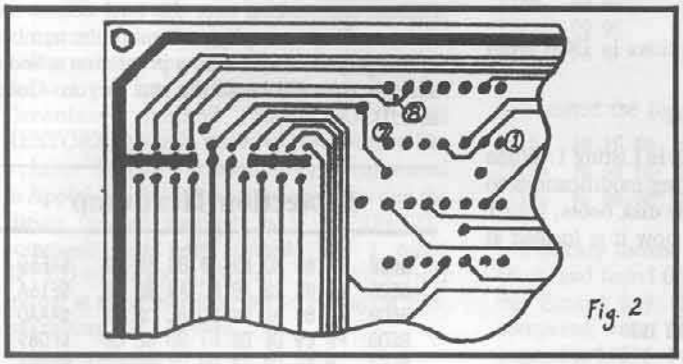


Fig. 2

and this type of modification will allow quick restoration of the card to its original condition.

Another alternative is to modify an IC socket instead of the IC itself, that is to bend out pins 6, 8 and 9 on the socket and proceed to solder the leads to it instead of the IC. If you use this method be sure to choose a socket that will plug into the socket on the card without problems. Simply plug the IC into the modified socket and then plug this assembly into the socket on the controller card. Removing the modified socket and replacing the IC will return your card to its original position.

Final Options

If you choose not to mount the switch I would recommend looping the leads through the small hole in the card at the back to act as a stress relief. If you do mount the switch somewhere on the case of your computer (mine is above and to the right of the keyboard), you can add a couple of small LEDs for a visual indication of which drive is sensed as drive 1. A 3PDT switch must be used to switch the LEDs. I have a green indication for a drive 1 boot and red for drive 2. I also have a 8-pin header/socket unit to make the connection between the switch and the card so that the card and the case aren't wired together.

Uses

Being able to reverse your drives may seem just a convenience at first but will soon become a necessity. You can have your adventure game in one drive and the game save disk in the other and simply flip the switch to access each one instead of swapping disks. The same applies to other programs that require changing disks.

I use mine consistently when trying to deprotect software. I keep a utility disk in drive 2 and the programs are always available. Some of the older copy programs don't allow you to choose the source and target drives; now you can.

I don't know if this has any practical use but here is an interesting trick you can do with this modification. Boot up your favorite nibble copier and put a blank disk in drive 2. Now position the head to a track on the blank disk by doing a data scan or whatever feature you have that allows you to do this. Let's say we have placed the head on track $8. Now go into the copy mode, select drive 1 as the source and target drive and read a track, say $11. When you are prompted to insert the copy disk flip the Drive Swapper switch and continue as if you had changed disks. Since the copier assumes that the same drive is being used to write and that the head is still in the proper position it will write to track $8 on drive 2, thus allowing you to move a track. You now have the catalog on track $8 but with all the markers of $11. Interesting.

After using the drive swapper switch for a while you'll wonder how you ever got along without it.

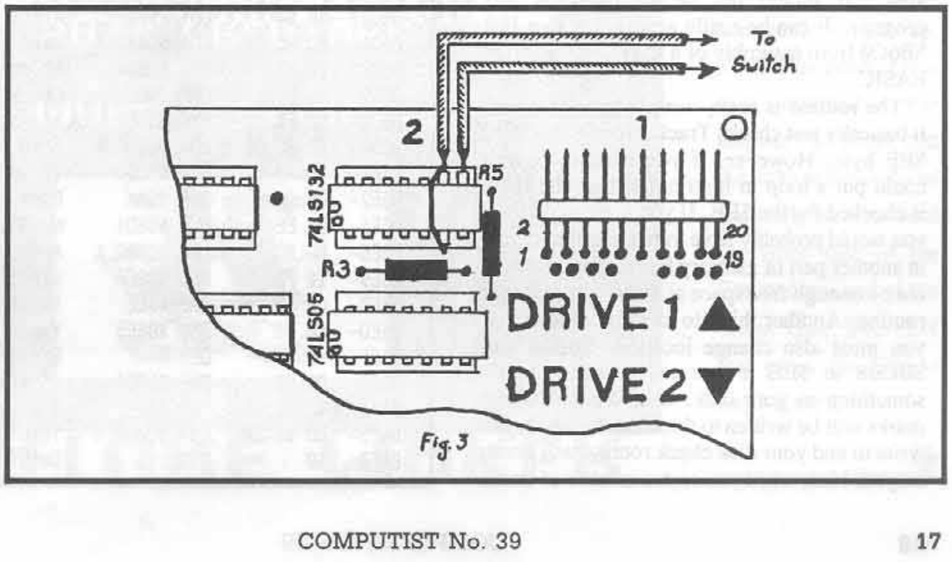


Fig. 3

copy protection:...

Abusing the Epilogues

by Rich Etarip

Quite some time ago I came up with a tricky method of copy protecting a disk which is kind of unique in a way. The disk will be easily copyable by any copier, but, there is a trick to it. By writing and calling a disk access routine, the computer will hang if the original disk is not in the drive. This is enough to defeat copying by the average user. Keep in mind though, that this method will not protect against bit copiers such as Locksmith. However, it will protect the disk from being easily copied. I like it because the person copying the disk is fooled into thinking that it is easily copied and later finds out that it is not. Here is an explanation of how it is done:

If you have been keeping up with COMPUTIST magazine, then you should know that DOS writes address and data markers to the disk to tell it where the data begins and ends on a disk. The standard end of address and end of data marks are DE AA EB. However, only the DE and the AA are used by DOS. The EB is not used and really has no effect on the disk if it is changed. Since normal DOS copiers do not copy the EB mark, if we were to change it to, say an EE, then check the disk to make sure the EE is present, a copy of the disk would not work. All you have to do is change locations $B8A8 and $BCB8 from $EB to $EE and initialize your disk. Then write a disk check routine to verify that the $EE is present on the disk. See Listing 1 for the disk check routine that I came up with.

The reason I put this routine at $B6C0 is because that memory is available in the DOS area and should not be disturbed by any program. It can be easily accessed with a JSR $B6C0 from assembly or a CALL 46784 from BASIC.

The routine is pretty simple to understand. It basically just checks Track 0 for the modified $EE byte. However, if you wanted to, you could put a loop in it so more than one track is checked for the $EE. If you were to do that, you would probably have to relocate the routine in another part of memory because there would not be enough free space at $B6C0 for a larger routine. Another thing to keep in mind is that you must also change locations $B8A8 and $BCB8 to $EE if you ever want to write something to your disk. Otherwise $EB end marks will be written to those tracks which you write to and your disk check routine will bomb on you. Plus, when you scan a number of tracks at a time, the drive head will move from track to track quickly and make a suspicious sound that would indicate the disk is being checked.

The best time to call this routine is right before or right after the disk is accessed because then it is not so obvious that you are just doing a check to the disk.

Making Your Own Protected Disk

1) The first step is booting a fresh DOS and entering the monitor.

```
PR#6
CALL -151
```

2) Next, change two locations in DOS from $EB to $EE.

```
B8A8:EE N BCB8:EE
```

3) Now enter the hex dump in Listing 1. When finished, make the following modifications to Boot 1 at $B600. When the disk boots, boot 1 loads into $800 but right now it is located at $B600.

```
B64A:4C B4 B6
B6B4:20 C0 B6 4C 00 B7
```

What these modifications will do is cause Boot 1 to call the disk access routine at $B6C0. That way, a copy of your disk will not even boot.

4) The next step is initializing your disk.

```
INIT HELLO
```

5) When the initialization is complete, reboot DOS (not with the disk you just formatted) and run a normal DOS copier such as COPYA and make a copy of your disk. When finished, try booting the copy. If all has been done correctly, the copy should just hang with the drive spinning. Pretty neat, huh?

It is a good idea to call the disk access routine a number of times from your program and you may even want to go as far as checking the disk access routine from your program to make sure that it has not been tampered with. This is easily accomplished by adding up all the bytes in the routine and making sure the total comes out correct. There is really no limit to the number of things you can do with a protection scheme such as this. I'll leave the rest to you. Good luck!!!

Protection Hexdump

```
B6C0: A2 60 BD 89 C0 BD 8E C0   $4E59
B6C8: A9 00 20 A8 FC A9 00 20   $E15A
B6D0: A0 B9 A0 04 BD 8C C0 10   $AAAD
B6D8: FB C9 DE D0 F7 BD 8C C0   $FD89
B6E0: 10 FB C9 AA D0 EE BD 8C   $E00A
B6E8: C0 10 FB C9 EE F0 06 88   $8E33
B6F0: D0 E2 4C F2 B6 BD 88 C0   $F477
B6F8: 60                        $FF44
```

```
B6C0-  A2 60     LDX #$60     Specify slot 6.
B6C2-  BD 89 C0  LDA $C089,X  Turn on the disk drive.
B6C5-  BD 8E C0  LDA $C08E,X  Tell the disk drive we are reading.
B6C8-  A9 00     LDA #$00
B6CA-  20 A8 FC  JSR $FCA8    Do a quick delay to allow drive to turn on.
B6CD-  A9 00     LDA #$00     Load accumulator with $00 for SEEK command.
B6CF-  20 A0 B9  JSR $B9A0    Call the SEEK command in DOS to seek Track 0.
B6D2-  A0 04     LDY #$04     Allow for 4 possible errors before hanging.
B6D4-  BD 8C C0  LDA $C08C,X  Read a byte from the disk
B6D7-  10 FB     BPL $B6D4    If not a valid disk byte, try again.
B6D9-  C9 DE     CMP #$DE     Check if byte is $DE.
B6DB-  D0 F7     BNE $B6D4    If not, get another byte.
B6DD-  BD 8C C0  LDA $C08C,X  A $DE was found, read another byte.
B6E0-  10 FB     BPL $B6DD    Valid byte?
B6E2-  C9 AA     CMP #$AA     Does $AA follow the $DE?
B6E4-  D0 EE     BNE $B6D4    NO.  Look for another $DE.
B6E6-  BD 8C C0  LDA $C08C,X  A $DE $AA pair was found.  Read the next byte.
B6E9-  10 FB     BPL $B6E6    Valid byte?
B6EB-  C9 EE     CMP #$EE     Is it our modified $EE?
B6ED-  F0 06     BEQ $B6F5    Yes!  Original disk present.  Go to end.
B6EF-  88        DEY          Error encountered.  Byte was not $EE.
B6F0-  D0 E2     BNE $B6D4    If 4 errors have not occured, try again.
B6F2-  4C F2 B6  JMP $B6F2    4 errors.  No $EE bytes found.  Computer hangs.
B6F5-  BD 88 C0  LDA $C0E8,X  Turn off disk drive.
B6F8-  60        RTS          Return from subroutine.
```

print shop companion's...

Driver Game

by Marc Batchelor

Broderbund Software Inc.
17 Paul Drive
San Rafael, CA 94903
$39.95

Requirements:
Print Shop Companion
64K][Plus, //e, //c

When I buy a program, I like to find out just exactly what makes it tick. So, I fool around a lot with the program (in its unprotected state thanks to COMPUTIST). The Print Shop Companion is no exception. Upon trying to boot up the back side of the disk, I took notice of the nice screen format and decided I might want to use this type of format later in my programs, so I pressed the Reset key on my //c, and the disk promptly rebooted. I then decided to boot the back side into auxiliary memory using Ken Greenlaw's method (XFER.BOOT and RESTORE) given in issue No. 16. Worked like a charm. Sort of. I pressed Reset and appeared in Applesoft. But, the strange part was that the cursor seemed sluggish as if a "SPEED" command had been issued. So, I typed SPEED=255 and hit Return. I found myself staring at the contents of memory location $B3. Interesting. So, I typed

3F8:4C ØØ Ø3

(which hooks up the CTRL Y vector), and BLOADed RESTORE. I then moved $8ØØ-$95FF from auxiliary memory to main memory with:

8ØØ<8ØØ.95FF CTRL Y

Then I did a massive BSAVE after I told DOS I could do so (A964:FF). I then rebooted DOS and BLOADed the file. I found that if I typed "1ØØØG" that neato-keen screen would pop right up. So I pressed Reset to disassemble the program and I got that sluggish cursor again. Grr. This time when I typed SPEED=255 everything went normally and I got my normally prompt prompt (pun intended). I was merrily listing from $1ØØØ on and I came upon some strange code starting at $119E.

```
119E-  C9 9E     CMP #$9B
11AØ-  FØ 6F     BEQ $11AD
11A2-  C9 9E     CMP #$9E
11A4-  FØ 6F     BEQ $1215
```

I started the listing at $1215 and got...

```
1215-  2Ø 2F FB  JSR $FB2F
1218-  2Ø 58 FC  JSR $FC58
121B-  4C ØØ 6Ø  JMP $6ØØØ
```

I quickly looked at my Beagle Bros ASCII chart, and found that this code compared with the Escape key ($9B), branched back and compared with "CTRL 6" ($9E), and if equal, jumped to $1215. I then powered down, booted up the back side of my disk, and when the message came up, I pressed (Escape)-CTRL 6. Who would have believed it? I was at the start of a game by Roland Gustafsson called DRIVER. It listed several skill levels, and a place that showed high scores. I guess the folks at Electronic Arts aren't the only ones who put undocumented features into their programs. To save this game into a file was then very easy. I took out my disk with the 131-sector chunk of memory that I had saved, BLOADed that chunk, and looked at $1215. Since $6ØØØ is an obvious place to put a program, I typed **6ØØØG**. Sure enough, there was the game. I pressed Reset and set my speed back to normal and took a look at the program. As it turns out, it was less than $5ØØ bytes long. I saved the game with:

BSAVE DRIVER,A$6ØØØ,L$48A

The game only took up 6 sectors, and worked like a charm.

In a nutshell:

1) Boot the back side of Print Shop Companion.

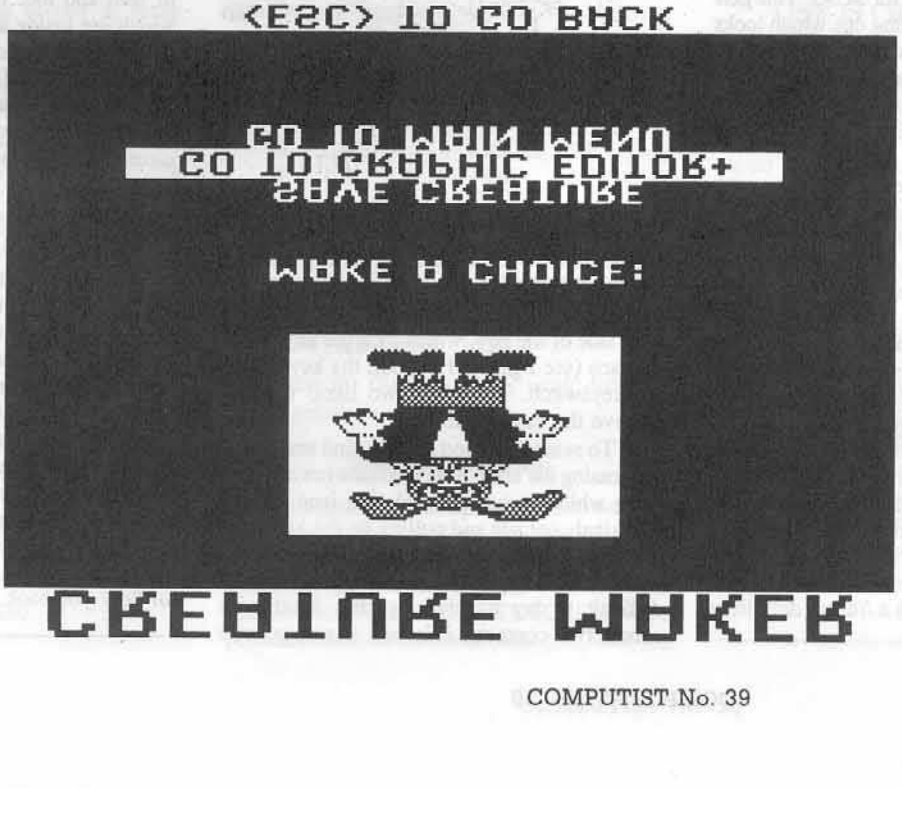
2) Take the Companion disk out of the drive and replace it with a slave disk with no HELLO.

3) Press Reset. It will reboot and you will find yourself in Applesoft.

4) Save the game to disk.

BSAVE DRIVER ,A$6ØØØ,L$48A

That's it. The game is rather lame, but fun. To change gears, press the desired gear number. To prove that the folks at Broderbund are really bizarre, boot the front side of the Companion, and at the menu type STEVEN and press the Escape Key. Have fun.



```
<ESC> TO GO BACK

GO TO MAIN MENU
GO TO GRAPHIC EDITOR+
SAVE CREATURE

MAKE A CHOICE:

CREATURE MAKER
```

Using a TI/99 keyboard for...

Keyboard Repair

by Philip Goetz

Warning: The procedure described below details modifications to your computer which could cause damage to it. COMPUTIST or SoftKey publishing will not be held responsible for any events that occur while following this article.

Requirements:
Soldering Iron
TI/99 Keyboard
Screwdriver
Desoldering tool
Vise Grips (optional)

Eventually, a key on your keyboard will stop working. It's happened to me five times. The symptom is simple; you press the key, but nothing happens. Your Apple dealer will be happy to replace your keyboard for, say, $145. But there's a cheaper solution.

All five times one of keys stopped working, it wasn't because my whole keyboard was broken, just the keyswitch for one key. A keyswitch electrically connects two solder terminals on its bottom when its key is pressed. You can buy keyswitches from some Apple dealers for about $5 each (be sure to specify which type of Apple you have). A much better deal is to buy Radio Shack's TI 99/4 keyboard (catalog number 277-1017) for $2.95. This gets you 48 keyswitches, including one which locks on or off. If you buy this, you will have to carefully take each keyswitch out of the keyboard as you use it, and save the plastic stem from your old Apple keyswitch, which is angled differently.

The procedure involves removing your Apple's keyboard, removing the new keyswitch from the other keyboard (unless you buy individual keyswitches), removing the old broken keyswitch from your Apple keyboard, putting the Apple keystem in the Radio Shack keyswitch (if you have those or some other keyswitch with a different stem), soldering the new keyswitch into your Apple keyboard, and putting your Apple back together. This will void your warranty, so don't tell anybody. Neither I nor COMPUTIST are responsible for any damages caused to anything or anybody by following, reading, or even thinking about this article. This article was written using an Apple][Plus and a clone; details (such as screw placement) may differ with a //e and definitely will with a //c.

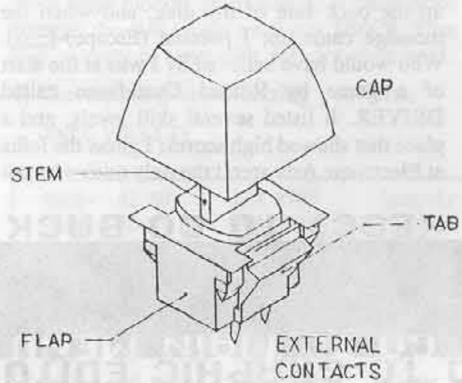
Removing the Keyboard

Plug in your soldering iron. Turn your Apple off. Unplug it at the wall and then pull the plug out of the computer. Disconnect your TV or monitor at the plug on your RF modulator or at the Video Out port, respectively. It may help to remove any disk drives at this point. Turn the computer upside-down, with the keyboard at the front.

Remove the four screws at the front, the two on each side, and the two farthest to the back. There is a third screw about 3/4'' forward and to the right of the left rear screw. Do not remove it; it holds the motherboard on. Put all the screws in a safe place.

Now slowly lift the bottom steel plate which holds the motherboard and power supply off of the case until you can unplug the keyboard cable from the motherboard. Draw a diagram showing which way the cable is attached before you remove it.

Set the case upside down and remove the six screws holding the keyboard on. Remove the keyboard.



Removing Keyswitches

A keyswitch is connected to the keyboard by two electrical contacts soldered to a printed circuit board, and by two plastic tabs, one on each side of the key, which lock the keyswitch in place (see figure). First pull the keycap off the keyswitch. There are two basic ways to remove the keyswitch:

1) To remove a good switch, find some way of pressing the ends of the two tabs toward each other while applying a soldering iron to both electrical contacts and pulling on the keystem. Since the tabs and the contacts are on opposite sides of the keyboard, and it is sometimes difficult to lay a soldering iron lengthwise across two contacts, it is best if you get one person to concentrate on pressing the tabs and pulling the key while you hold the board and melt the solder. You may want to apply pressure on the contacts with the soldering iron to help push the keyswitch out. I use needlenose pliers to press the tabs and a vise grip on the keystem to pull the keyswitch out. Both are done on the top side of the keybord. I'm sure there are better ways. With patience, the switch will eventually come out.

2) To remove a bad switch, munge it (mash until no good). Get all your small, sharp, gripping and tearing tools and rip the switch into pieces. If your replacement switch has a different-looking keystem (i.e. Radio Shack), you must save the Apple keystem (see figure); be careful not to damage it. It is a good idea to have someone hold a vacuum cleaner near the keyswitch once you have the keystem out to keep shards of plastic out of the keyboard.

Replacing the Keystem

Once you have both the good and bad keyswitches out, put the old keystem in the new keyswitch if necessary. To do this, you must first pry the black plastic top half of the keyswitch case off of the bottom half. Do not try to pull out the 4 black wedge-shaped tabs which are part of the top half. Instead, pull the 2 grey flaps from the bottom half out of their way. Fingernails may work if you haven't cut them lately.

After removing the top half, be careful not to spill and lose the spring, stem, or contact which are inside it. Replace the keystem with the old one, and carefully align the stem and vertical contacts (parallel to the stem) so you can slide the top half of the keyswitch case back on. You may want to test the keyswitch electrically before soldering it into your keyboard.

Putting it All Together

Put the good keyswitch into the hole where the bad one used to be. It is impossible to put it in upside-down because the external contacts are both on one side. It may not slide in all the way because hardened solder prevents the outside contact points from going through the two little holes in the keyboard's printed circuit board. If so, encourage it with a soldering iron. Snap the keyswitch in place and glob some solder over the contacts. Put the keycap onto the keystem. Finally, unplug your soldering iron and put your Apple back together opposite of how you took it apart.

A //c fix for the... Applesoft Sampler

by Jerry D. Greer

Requirements:
The Applesoft Tutorial
The Applesoft Sampler Disk
Apple //c Computer

When a friend of mine got away from Apple computing, (hard to believe but it happened), she gave me her copy of the Applesoft Tutorial with the Applesoft Sampler Disk that comes with it. She was moving into more advanced languages using mainframe computers.

On the other hand, I had just acquired my Apple //c and was anxious to see what it (and I) could do. I soon discovered that the sampler disk was not compatible with the //c. It was obviously not intended to be used on the new //c machines.

For many, it may not be worth worrying about. For others, the routines supplied on this disk are helpful and easy to use. The programs come in very handy for some simple but significant programming tasks. The "MAGIC MENU" is one of the these special routines. It has shown up in several of my projects. The disk (as expected) works fine on the //e. However, the programs contain a screen formatting routine (using some memory peeks). The problem for the //c is in one of these routines.

Identifying the Problem

If you have tried to run the "Magic Menu" on a //c, you have seen the problem. When the main menu for the sampler comes up, it is noticeable that the menu topics do not line up with their item numbers. The column of numbers is shifted down one full line leaving the topic headings off by a line.

Despite this offset, the item numbers still refer to specific programs on this menu (there are 4 of them). If you select number 1, EXAMPLE PROGRAMS FROM THE APPLESOFT TUTORIAL, you will watch as the numbers and topics for this next menu section roll past. Each topic is printed to the same, top line on the screen. The final line shows this:

```
      RETURN TO MENU
12.
```

Again, note that the topic line is off by one line from the item number. And again, even though the numbers and topics have disappeared from the screen, you can still select any number from 1 through 12 and the sample program to which it refers will still run. The trouble is that you have no list for reference when selecting! By looking over all of the problem areas, it becomes obvious that the program examples will run but the menus do not act right.

The Source of the Problem

After you have identified the problem, this step is relatively easy. In fact, you will find the BASIC program listed in Appendix E of the Applesoft Tutorial manual for Magic Menu, Disk Menu, and Converter. Looking at those listings, you will notice that one block of the program is devoted to a "menu maker". I looked through the listing to see if I could find some code that might be the source of the problem.

Because the problem concerned scrolling and the alignment of the item numbers with the topics, line number 625 in the MENU MAKER routine caught my attention. To see how it looks, list this line as it appears in the program. Get into BASIC and do this:

LOAD MAGIC MENU
LIST 625

The line will appear something like this:

```
625 VTAB PEEK (37): IF COL80
  THEN VTAB PEEK (1531): REM
  BELOW PRINT MOVES DOWN 1 LINE:
  THIS COMMAND IS A TRICK TO MOVE
  UP EXACTLY 1 LINE FIRST
```

This line in the program causes the print on the screen to move up one line while the last line is printed, in turn, onto the screen display. So, I simply rewrote line 625 to be like this:

625 VTAB PEEK (1531)

and wrote the file back to my work disk (not the original).

Test the Solution

At this time, I booted up the disk and watched the menus come up. They behaved like they were supposed to this time. All of the 12 menu selections on the EXAMPLE PROGRAMS FROM THE APPLESOFT TUTORIAL were properly arrayed on the screen.

Fixing the Program for the //c

Get into BASIC. The Applesoft prompt "]" will appear in the lower left corner of the screen. Insert the Applesoft Sampler disk in the drive and follow these steps to make the fix on each of the three programs that are giving you troubles.

A) First, put the magic back into the MAGIC MENU.

UNLOCK MAGIC MENU
LOAD MAGIC MENU
625 VTAB PEEK (1531)
SAVE MAGIC MENU
LOCK MAGIC MENU

B) Second, put the display back into the DISK MENU.

UNLOCK DISK MENU
LOAD DISK MENU
625 VTAB PEEK (1531)
SAVE DISK MENU
LOCK DISK MENU

C) Third, convert the CONVERTER.

UNLOCK CONVERTER
LOAD CONVERTER
625 VTAB PEEK (1531)
SAVE CONVERTER
LOCK CONVERTER

Final Touches

After you have made the changes in the program so that you can use it on the //c, label the disk so that it will not be confused with the other, original. It would also be a good idea to post some note about the change in the program. Instead of the brief entry I have shown above, you might like to do something like this:

625 VTAB PEEK (1531): REM THIS LINE IS CHANGED FROM THE ORIGINAL FOR //c USE

Notice that it is permissible for you to make changes in this program for your own use. In Appendix E of the Tutorial manual, one of the stated objectives is "To offer useful routines that you can incorporate into your own programs." Another objective states in part that "... The programs discussed here are to be played with, modified, experimented with, and changed."

The entire Tutorial applies equally well to users with the //c computer but a few things have to be disregarded. For instance, you should ignore all references to using a cassette recorder. A few other very minor differences will show up if you go through the lessons but they are not critical. The program is worth having if you are learning to program in BASIC. We owe a vote of thanks to Apple Computer for giving us this program to play with!

softkey for...

MIDI/8 Plus

by Ray Darrah

Passport Designs Inc.
625 Miramontes St.
Half Moon Bay, CA 94019
(415) 726-0280

Requirements:
Locksmith Fast Copy or similar
Sector editor or Copy][Plus

It looks like the folks at Passport have done it again. It seems they didn't heed my intense dislike for the copy protection on their MIDI/4 program (article appearing in COMPUTIST No. 28). I was hoping that MIDI/8 Plus would be distributed without copy protection. So much for hope.

As I suspected, the protection on this disk was similar to MIDI/4's protection. They are still using a file called READER to perform a nibble count on tracks $03 and $10. If the nibble count fails four times, then the computer locks up and beeps at you until you shut it off.

Luckily, getting around the protection is also accomplished in a similar manner. It involves re-arranging the boot sequence so that the READER file is loaded but never executed.

The first step in re-arranging the boot sequence is to figure out what the current sequence is. One way to do this is to insert MONCIOs here and there throughout the boot. Since the program uses only a slightly modified version of Diversi DOS, another way to examine the boot sequence is to use an NMI card (like my handy Abacus Know Drive) to stop the program for a moment and change location $AA5E to a $40.

Doing this revealed that the DOS commands were issued in the following order:

```
BRUN TABLES
BLOAD READER
BRUN DDM
BLOAD PDIM8P1
BLOAD BRUNM8P+
BLOAD M8.OBJØ
BLOAD M8.OBJ1P
BLOAD CLK24TBL
BLOAD STK
BLOAD M8P2
```

There is a file on the disk called HELLO but it is a blank Applesoft file that is never executed. Copy][Plus quickly revealed that Passport chose TABLES as the boot file.

Since the READER file wasn't BRUN, it must be executed by some other program (most probably by TABLES). A little closer examination revealed that READER was BLOADed at $8DØØ and at $98C (in the TABLES program) there was a JMP to $8DØØ.

Apparently, if everything went fine on the nibble count, the READER file would then BRUN DDM, BLOAD PDIM8P1 (the title picture), BLOAD BRUNM8P+ and executed it. Due to the *extreme* perversion of the code in the READER file, I was unable to find the code that actually did this. But it seems logical.

A little testing revealed that the BRUNM8P+ file loaded the remaining files, did a little memory setup and returned. A quick stroll down memory lane at this time (and a little Know Drive snooping) showed that the original program would execute fine from this point with an AØ1G from the monitor.

The Patching

First of all, I copied the original disk with Locksmith Fast Copy and ignored the expected errors on tracks $03 and $1Ø. Then I changed the boot program to be the usual HELLO instead of TABLES. Next, I wrote a short BASIC program that did the work that READER file does if everything goes well and BRUNs TABLES when finished. Then I patched TABLES so that it calls BRUNM8P+ and jumps to $AØ1 to start up the program.

After all that work, it bombed into the monitor. The area around $AØ1 was garbage on the copy but when I booted the original there was valid code there. I then noticed that if you EORed every byte from $AØ1 to $BØ1 with $Ø9 the code would be the same as the original disk's.

I therefore extended my patch to TABLES to include the EORing of all the bytes from $AØ1 to $BØ1 and voila, a perfect deprotected copy of MIDI/8 Plus! Hooray.

Step by Step

1) Copy MIDI/8 Plus with Locksmith fast copier or any whole disk copier that can ignore errors on tracks $3 and $1Ø.

2) Change the boot program to be "HELLO" instead of tables. This can be accomplished in one of two ways:

a) Use the Copy][Plus "Change Boot Program" option.

b) Use a sector editor and edit track $Ø1, sector $Ø9. Starting at byte $75, replace the word TABLES with the word HELLO. Next, edit track $ØØ, sector $ØD byte $42 from a $34 to a $Ø6.

3) Boot normal DOS, insert the copy of MIDI/8 Plus and type the following:

FP

```
1Ø PRINT CHR$(4)"BRUN DDM"
2Ø HGR2:PRINT CHR$(4)"BLOAD PDIM8P1"
3Ø PRINT CHR$(4)"BLOAD BRUNM8P+"
4Ø PRINT CHR$(4)"BRUN TABLES"
```

UNLOCK HELLO
SAVE HELLO
LOCK HELLO

4) Now we will change the TABLES file so that it won't execute READER but instead, call BRUNM8P+, EOR $AØ1 - $BØ1 with $Ø9 and start the program.

FP
BLOAD TABLES
CALL -151

```
98C:2Ø ØØ 8Ø A2
99Ø:ØØ BD ØØ ØA 49 Ø9 9D ØØ
998:ØA E8 DØ F5 AD ØØ ØB 49
9AØ:Ø9 8D ØØ ØB 4C Ø1 ØA
```

UNLOCK TABLES
BSAVE TABLES,A$8Ø1,L$1A6
LOCK TABLES

5) Put your original disk in a death star far, far away and make as many backup copies of the MIDI/8 Plus as you wish.

Thanks to Ray Feruski

softkey for...

Homeword v2.1

by The Nipper

Sierra On-Line
Coarsegold, CA 93614

Requirements:
48K Apple
COPYA
Sector editor

I know that there has been a crack around for the original version of Homeword for a while (Book of Softkeys 1) but as any of you who own the newer 2.1 version know that that crack doesn't work for this version. Although this version takes longer to boot due to the loading of two hi-res pages, it does support a larger number of printer cards including the //c. Since I sometimes use my Homeword to edit my source code I decided to open it up.

The Protection

A quick look at the Homeword disk with the nibble editor from my Copy II Plus (you can use any nibble editor) showed me that the disk had a standard DOS 3.3 format. I hauled out my COPYA and made a copy. When I tried to boot the copy, the disk booted as usual through the first two hi-res screens and then scrambled what was on the hi-res and shut off the drive. By watching the boot sequence on my Track Star, I could see that the trouble occurred after the drive arm swung back to track $Ø. Aha! The famous Sierra On-Line nibble count.

I then rebooted my copy and stopped it when it swung to do the check of track $Ø. Bingo, I found myself staring at some fascinating code on the $2EØØ page of memory. A quick check of the disk found this code on track $1Ø sector $9. This code looks for a sequence of bytes on track $Ø. The code that is important to us starts at $2E57 and looks as follows:

```
2E57-  LDA $CØ89,X     turn on drive
2E5A-  LDA #$Ø5
2E5C-  STA $BBØ2       put a $Ø5 at $BBØ2
2E5F-  JSR $2E93
2E62-  BPL $2E65       always taken
2E64-  JSR $CØC8
```

The first instructions turn on the drive and then put a $Ø5 in location $BBØ2. This location is the key since any subsequent error that is encountered in the check of track $Ø decrements $BBØ2. If you check the boot with your original disk you will find that a $Ø5 is still in this location after the check of track $Ø has occurred. Now the JSR that follows goes to the subroutine at $2E93, which finds sector $Ø and reads the address information and data prologue. It then reads in $FF bytes plus $5B of data ($15A) and returns. Since there are $156 bytes in a data field plus one checksum byte this routine reads to the end of the data epilogue bytes. The BPL (Branch if PLus) at $2E62 is taken into the middle of the next instruction yielding the following code:

```
2E65-  INY              increment counter
2E66-  CPY #$3Ø         check if y=$3Ø yet
2E68-  EOR $CØ8C,X      read a byte
2E6B-  BCC $2E65        loop until y=$3Ø
2E6D-  LDA $CØ8C,X      read a byte
2E7Ø-  BPL $2E7C        wait until legal byte
2E72-  CMP #$C9         is it $C9?
2E74-  BNE $2E83        indicate error if not
2E76-  LDA $CØ88,X      turn off drive
2E79-  JMP $2E8B        exit routine
2E7C-  NOP
2E7D-  NOP
2E7E-  LDA $CØ8C,X      read byte
2E81-  BMI $2E72        branch if valid byte
2E83-  DEC $BBØ2        else decrement count
2E86-  BNE $2E5F        retry if not zero yet
2E88-  JMP $2E76        drive off and exit
2E8B-  LDA $BBØ2        get error byte
2E8E-  STA $Ø8          store it
2E9Ø-  STA $Ø9
2E92-  RTS              return to program
```

Now the code from $2E65 to $2E6B reads in an additional $3Ø bytes. If you look at the raw data on track $Ø with a nibble editor you will see that the sync field between sectors $Ø and $1 is very large and instead of being full of $FFs it has a wide variety of illegal bytes including $C9s. The code from $2E69 to $2E74 checks to see if the next byte is a $C9. If it isn't it goes to $2EB3 which decrements $BBØ2. It then checks to see if $BBØ2 is zero yet and starts over again if not. If it is then it jumps back to $27E6 which is where we would have been if we had found the $C9. The drive is turned off and we jump to $2E8b. Here the value in $BBØ2 is put in two zero page locations and we go back to the main program. So, if the $C9 wasn't found, our zero page locations would contain $ØØ and would contain a greater value if it was.

So what we have is not a nibble count but a routine that looks for an illegal byte in the sync field between sectors $ØØ and $Ø1 on track $Ø. Since a COPYA of our original disk will have cleaned up the sync field with $FFs, we know the CoMPare at $2E72 will be negative. All we need to do is disable it. My first attempt was to change the bytes at $27E4 from $DØ ØD (BNE $2E83) to $DØ ØØ (BNE $2E76). This causes the code to continue on to the next instruction regardless of the outcome of the CoMPare. Unfortunately, when I went to boot the disk the boot stopped before it even got to the check of track $Ø.

This usually means that there is a checksum being done elsewhere in the program to insure that the code which checks track $Ø has not been altered. The first thing that I tried was to see if I could find the jump subroutine that called the code in the hope that I could trace it back to the checksum routine. However, I couldn't find the JSR and decided that it was being done indirectly through the stack, so I gave up on that idea. The other possibility is to disable the branch at $27E4 while leaving the checksum unaffected. Checksums are done with EORs (Exclusive OR), and as any of you have spent time with a piece of paper working out EORs know the result is not affected by the order in which the bytes are EORed. That is, a group of ten numbers will always produce the same EOR result regardless of which number is EORed first, second, etc. Luckily there are two NOPs ($EA, No OPeration) in the code at $2E7C which will not be used on the check of our copy, so it is simply a matter of switching them with the instruction at $2E74.

The Fix

Use COPYA to copy your Homeword disk. Boot up your sector editor and make the following changes:

Track	Sector	Byte	From	To
$1Ø	$9	$74	$DØ	$EA
		$75	$ØD	$EA
		$7C	$EA	$DØ
		$7D	$EA	$ØD

That's it! you now have a COPYAable version of your Homeword version 2.1.

Borrowed Time

by Jeanne Edwards

Activision
PO Box 7287
Mountain View, CA 94039
$39.95

Requirements:
Apple][
Super IOB 1.5
A sector editor
A disk search utility
A notched double sided disk

Whether you are into adventure games or not, I believe you'll find **Borrowed Time** (BT) from Activision is by far a superb combination of graphics and technical expertise. With the use of pull-down menus and a list of possible commands, BT is a very enjoyable game requiring more than the usual amount of skill in solving the mystery. You take on the role as Private Eye Sam Spade with a cast of many, and attempt to solve the case. Meanwhile, a couple of no-good thugs are trying to retire you early, permanently, at every move. The authors have put a lot of effort into this adventure to take into account the myriad of commands that the player might issue. The only playing tip I have at this time is to issue "QUICKSAVE" often. Because the game writes to the master disk and is copy protected, I felt it needed to be deprotected.

This was my first effort at creating a controller for Super IOB. At first, I thought it would be too complicated, but after some delving into some sample controllers and reading the documentation I was quite impressed at the ease of use as well as the capabilities it provides.

In scoping out BT to find out how to escape the killers (the copy protectors), I was unable to read the disk. So what's a person to do? Call the Ghostbusters? No, Activision has copy protected that also. I have almost all of the back issues of COMPUTIST and they have proved to be invaluable in deprotecting disks. Issue No. 3Ø has a softkey for "Space Shuttle" by Activision but it won't work on this game. Do you think the software companies have a subscription to COMPUTIST? I just love a challenge, so let's put on our eye patches and brandish our rapiers while the timid at heart go straight to the cookbook solution.

First we need to boot up our nibble editor. I use Copy II Plus 6.Ø so I can switch back and forth bewtween the nibble editor and sector editor without having to reboot. The address field prologue for track Ø is normal, but the address field epilogue changes for each sector throughout the disk. From track Ø to $11, they use different address prologues and then start over with the same sequence again all the way through to track $22. The data on the reverse side is protected in the same manner so this softkey needs to be used for both sides.

Here is a list of the address prologues used on each track:

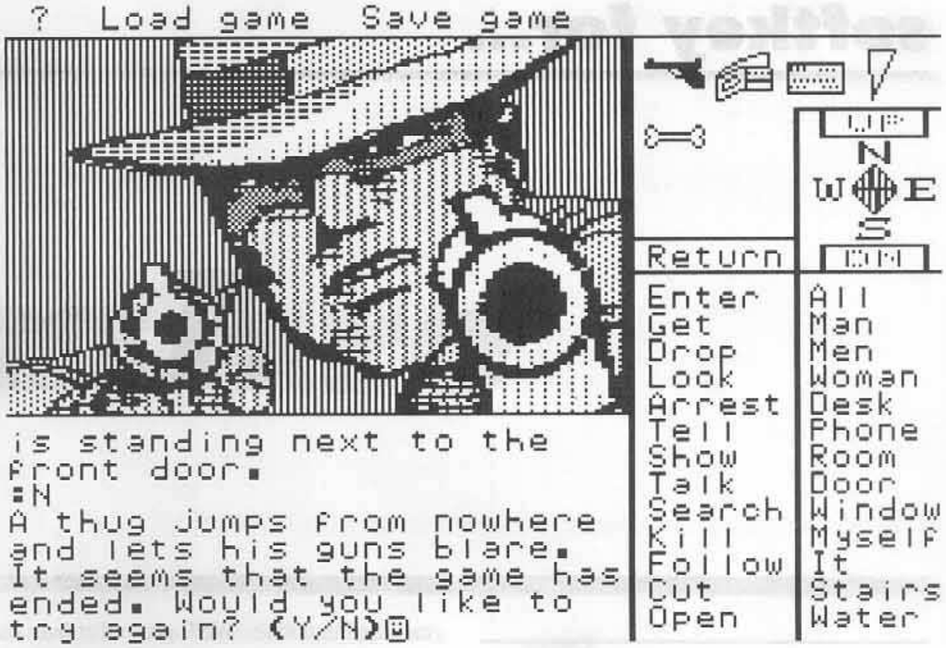
Track	Address Prologue
$Ø	D5 AA 96
$1,$11,$21	D5 97 EE
$2,$12,$22	D5 9A EF
$3,$13	D5 9B F2
$4,$14	D5 9D F3
$5,$15	D5 9E F4
$6,$16	D5 9F FF
$7,$17	D5 A6 F7
$8,$18	D5 ED 96
$9,$19	D5 EE A6
$A,$1A	D5 EF AA
$B,$1B	D5 F2 D5
$C,$1C	D5 F3 DF
$D,$1D	D5 F4 EA
$E,$1E	D5 F5 AE
$F,$1F	D5 F6 FE
$1Ø,$2Ø	D5 96 ED

While making the Super IOB controller, I noticed that Super IOB has the capability to alter the address epilogues but not to ignore them. This can be done with a POKE 475Ø7,Ø and POKE 47517,Ø while reading, then while writing, change them back to normal with a 174 and a 164 respectively. This feature can be added to Super IOB itself as I have done, but for you readers, I created a controller to take care of ignoring the address epilogues while reading and restoring them back to normal when writing.

Once we have a deprotected version, there are a few sector edits to conduct. The address epilogue checks can be edited to read normal DOS 3.3 epilogues (DE AA) by changing the instructions that check for the protected ones. On track Ø sector 1, byte $FB, you will find D9 AD 11 (CMP $11AD,Y). Change this to C9 DE EA (CMP #$DE, NOP); the NOP is there to replace the third byte of the original instruction. At track Ø sector 2, byte $Ø6, change D9 BD 11 (CMP $11BD,Y) to C9 AA EA (CMP #$AA, NOP).

Borrowed Time's address prologue lookup table was found on track Ø sector 9, starting at byte $AD by using a disk search utility to look for a sequence of changed address prologues, i.e. 97 9A 9B (see above table). To fix this, change all bytes $AD-$BC to the value $AA and change bytes $BD-$CC to $96. At this point, I thought I was finished because I couldn't locate an address prologue table on side 2. After booting up side 1 of the deprotected version, side 2 would not be accepted, but the program would accept the protected side 2. Ah Watson, the address prologue table for side 2 is on side 1. Back to the disk searcher… at track 3, sector $D, addresses $5A-$69 and $6A-$79, I found it. That table is fixed the same way.

Additionally, the address epilogue compare instructions need to be taken care of at track 3 sector $B, byte $56, changed from D9 5A 6D (CMP $6D5A,Y) to C9 DE EA (CMP #$DE, NOP) and at byte $61, from D9 6A 6D (CMP $6D6A,Y) to C9 AA EA (CMP #$AA, NOP). Be sure to save the changes to the disk.



```
is standing next to the
front door.
:N
A thug jumps from nowhere
and lets his guns blare.
It seems that the game has
ended. Would you like to
try again? (Y/N)
```

Cookbook

1) Install the Borrowed Time controller in Super IOB. Line 1020 of the controller tells Super IOB to ignore the address epilogues while line 1060 normalizes these checks during the write sequence. Line 1050 checks for tracks 16 and 32, where the sequence of changed address prologues starts over. The data statements are RESTOREd rather than having to type in 34 data statements.

2) Copy both sides of Borrowed Time with the same controller.

3) Using your sector editor, perform the following sector edits on side 1 only.

Track	Sector	Byte(s)	From	To
$Ø	$1	$FB	$D9	$C9
$Ø	$1	$FC	$AD	$DE
$Ø	$1	$FD	$11	$EA
$Ø	$2	$Ø6	$D9	$C9
$Ø	$2	$Ø7	$BD	$AA
$Ø	$2	$Ø8	$11	$EA
$Ø	$9	$AD-$BC	$xx	$AA
$Ø	$9	$BD-$CC	$xx	$96
$3	$D	$5A-$69	$xx	$AA
$3	$D	$6A-$79	$xx	$96
$3	$B	$56	$D9	$C9
$3	$B	$57	$5A	$DE
$3	$B	$58	$6D	$EA
$3	$B	$61	$D9	$C9
$3	$B	$62	$6A	$AA
$3	$B	$63	$6D	$EA

4) Put away your master disk and start playing, because you're living on Borrowed Time.

controller

```
1000 REM BORROWED TIME
1010 TK = Ø : LT = 35 : ST = 15 : LS = 15 : CD = WR : FAST
     = 1
1020 GOSUB 490 : POKE 47507 ,Ø : POKE 47517 ,Ø : MB
     = 55 : IF TK > Ø THEN GOSUB 190
1030 GOSUB 610 : MB = MB + 16 : TK = TK + 1 : IF TK
     = 17 OR TK = 33 THEN RESTORE
1040 IF MB < 152 THEN GOSUB 190 : GOTO 1030
1050 GOSUB 230 : POKE 47507 ,174 : POKE 47517
     ,164 : TK = TK - 7 : MB = 151
1060 GOSUB 490 : GOSUB 610 : IF PEEK (TRK ) = LT
     THEN 1080
1070 TK = PEEK (TRK ) : ST = PEEK (SCT ) : GOTO 1020
1080 HOME : PRINT "COPYDONE" : END
5000 REM NO CHANGE ON TRK Ø
5010 DATA 213 ,151 ,238
5020 DATA 213 ,154 ,239
5030 DATA 213 ,155 ,242
5040 DATA 213 ,157 ,243
5050 DATA 213 ,158 ,244
5060 DATA 213 ,159 ,255
5070 DATA 213 ,166 ,247
5080 DATA 213 ,237 ,150
5090 DATA 213 ,238 ,166
5100 DATA 213 ,239 ,170
5110 DATA 213 ,242 ,213
5120 DATA 213 ,243 ,223
5130 DATA 213 ,244 ,234
5140 DATA 213 ,245 ,174
5150 DATA 213 ,246 ,254
5160 DATA 213 ,150 ,237
```

controller checksums

1000	- $356B	5040	- $1932
1010	- $2544	5050	- $8C1B
1020	- $F402	5060	- $62CB
1030	- $D637	5070	- $A1D9
1040	- $A189	5080	- $Ø13C
1050	- $957D	5090	- $8C54
1060	- $Ø711	5100	- $C782
1070	- $7CAA	5110	- $42BA
1080	- $AB6B	5120	- $DD92
5000	- $224F	5130	- $611B
5010	- $C591	5140	- $D2DB
5020	- $C9D5	5150	- $DB6F
5030	- $43A9	5160	- $3871

```
:GO STAND
"Hi, peeper, heard the
latest?" he asks.
You greet your pal,
Hawkeye, who runs New
City's most popular
newsstand.
```

Amazon

by Rich Etarip

Trillium Software
One Kendall Square
Cambridge, MA Ø2139

Requirements:
64K Apple II Plus or compatible
DOS 3.3 System Master disk
A sector editor
4 blank diskette sides
Amazon disks

Amazon is a four-sided hi-res adventure game from Trillium Software. I didn't get much of a chance to look at the game itself because I was really more interested in looking at the protection used on it. The first thing I found out is that two of the sides are protected the same way and two sides are not protected at all. Now you would think that because of this, the program would use a normal RWTS to read from the normal disks and a modified RWTS to read from the protected disks. Wrong. Amazon uses a very strange looking modified RWTS which is able to read from either of the disks depending on which it encounters. What this softkey will involve is first normalizing the two protected sides of the disk, then replacing their RWTS with a normal DOS 3.3 RWTS.

Before I could normalize the protected disks, I had to know how they were protected first. To find this out, I took out a nibble editor and examined the address and data marks on the disk. From this I found that Amazon uses modified address marks of AA D5 AB and DE AB and data marks of AA D5 EB and ED AA. What is involved here is reading from the original disks using their address and data marks and then writing to the copy disk using normal address and data marks. Super IOB can be made to do this easily, but just for kicks the COPYA program from the DOS 3.3 master disk can be modified to do this. We'll get to that later on.

1) One thing that is ESSENTIAL to the deprotection of this game is the volume numbers of the disks. Sides 1 through 4 MUST have the respective volume numbers of 1 through 4. This should be taken care of before anything else is done. You should also label each side so you know which side is which.

INITialize side 1 with volume number 1.

```
INIT HELLO,V1
```

Repeat with sides 2,3,and 4.

```
INIT HELLO,V2
INIT HELLO,V3
INIT HELLO,V4
```

2) Now that you are finished, LOAD COPYA from the system master diskette and change line 25Ø so it does not format your copy disks.

```
25Ø FT=1
```

After you make this change run the program and copy sides 2 and 4 of Amazon. COPYA should have no problem copying these sides because they are normal DOS.

3) Once the copying is finished make the following modifications to the COPYA program:

```
7Ø POKE 47426,24 : POKE 929,234
196 GOSUB 4ØØ
257 GOSUB 5ØØ
```

Also, enter lines 4ØØ through 55Ø from Listing 1. These lines modify DOS to read from the protected disks and write back to normal DOS. The numbers in the POKE statements must be entered exactly as seen or the program will not be able to copy so type carefully.

4) After you have added lines 4ØØ through 55Ø into COPYA (I know, I hate typing too) run the program and copy sides 1 and 3. The disk drive will do some grinding during the first read phase because it can't read track Ø, sector Ø (the only normal sector on the disk). However, the rest of the disk will copy with no problem.

5) Now that the copying is complete, reboot DOS and run your sector editor. Then read track Ø sector Ø from the original Side 3 and rewrite it to the copy Side 3. Now make the following sector edits to Side 3. Keep in mind that Side 3 is the Boot Side.

Track	Sector	Byte	From	To
Ø	Ø	4A	AA	D5
Ø	Ø	53	D5	AA
Ø	Ø	5D	AB	96
Ø	Ø	88	AA	D5
Ø	Ø	91	D5	AA
Ø	Ø	9B	EB	AD
Ø	3	1A	AA	D5
Ø	3	23	D5	AA
Ø	3	2D	AB	96
Ø	3	57	AA	D5
Ø	3	6Ø	D5	AA
Ø	3	6A	EB	AD

6) The final step in this softkey is replacing Trillium's RWTS with a normal RWTS. A problem arises here. Trillium's RWTS runs from $B6ØØ to $BDFF and the normal RWTS is at $B8ØØ to $BFFF. If we were to replace their RWTS with ours it would not work. However, we can relocate the normal RWTS to $B6ØØ. Apple's Programmer's Aid ROM contains a machine language relocating routine, so we can use this to relocate the RWTS at $B6ØØ.

The first part of this step is getting the Programmer's Aid ROM into memory. It is no longer available normally (as ROM) but Apple, in its infinite wisdom, includes it inside the Integer BASIC file on the DOS 3.3 System Master. Boot the DOS 3.3 Master diskette so that the HELLO program loads Integer into the RAM card. When this is finished, get into Integer BASIC and enter the monitor.

```
INT
CALL -151
```

Adventure

Now call up the Programmer's Aid Relocating routine at $D4D5.

D4D5G

To relocate a block of memory you must specify the destination address ($B6ØØ) followed by a less-than sign (<), and then the start and end address of the memory you want relocated (B8ØØ.BFFF). Finally type a ⌃Y, an Asterisk (*), and press Return.

B6ØØ<B8ØØ.BFFF⌃Y *

Now you must type the same exact thing again, but excluding the asterisk this time.

B6ØØ<B8ØØ.BFFF⌃Y

The relocated RWTS is at $B600 now and it must be moved to safe memory so we can reboot DOS. Move it to $46ØØ and then reboot a regular DOS.

46ØØ<B6ØØ.BDFFM
C6ØØG

7) Now all we have to do is write the relocated RWTS to Side 3 of Amazon. Enter the monitor first.

CALL -151

Then enter the hex dump in Listing 2. This routine will write the RWTS at $46ØØ to the Amazon disk. This will be written to Track Ø Sectors $8 through $F.

Once Listing 2 is entered, insert the backup of Amazon Side 3 and execute the routine at $6ØØØ.

6ØØØG

We're all finished. You should now have a working COPYAble version of Amazon. Remember, if you wish to make further backup copies of Amazon, the four sides must contain their respective volume numbers. The best copy program to use is the Locksmith quick copier because it copies the volume of the disk too.

Have fun!

```
K-337 AMAZON   landsat A-505
INPUT
PROCESSED
OUTPUT
trans = 21.09321  lat  = 09.57s  file-01
rcvd  = 29.02349  long = 76.30w  sctr-09
INTRUDER!  : gain-09.991
```

listing 1

```
400 REM CHANGE MARKS FOR READING
410 POKE 47335 ,170 : POKE 47345 ,213 : POKE
    47356 ,235 : POKE 47413 ,237 : POKE 47423
    ,170
420 POKE 47187 ,170 : POKE 47192 ,213 : POKE
    47197 ,235 : POKE 47262 ,237
430 POKE 47445 ,170 : POKE 47455 ,213 : POKE
    47466 ,171 : POKE 47505 ,222 : POKE 47515
    ,171
440 POKE 48250 ,170 : POKE 48255 ,213 : POKE
    48260 ,171 : POKE 48312 ,171
450 RETURN
500 REM CHANGE MARKS FOR WRITING
510 POKE 47335 ,213 : POKE 47345 ,170 : POKE
    47356 ,173 : POKE 47413 ,222 : POKE 47423
    ,170
520 POKE 47187 ,213 : POKE 47192 ,170 : POKE
    47197 ,173 : POKE 47262 ,222
530 POKE 47445 ,213 : POKE 47455 ,170 : POKE
    47466 ,150 : POKE 47505 ,222 : POKE 47515
    ,170
540 POKE 48250 ,213 : POKE 48255 ,170 : POKE
    48260 ,150 : POKE 48312 ,170
550 RETURN
```

listing 2

```
6ØØØ: A9 ØØ 8D EC B7 8D EB B7   $9297
6ØØ8: 8D FØ B7 A9 Ø8 8D ED B7   $DACB
6Ø1Ø: 8D FF B7 A9 46 8D F1 B7   $3EE6
6Ø18: A9 Ø2 8D F4 B7 A9 B7 AØ   $C551
6Ø2Ø: E8 2Ø B5 B7 EE F1 B7 EE   $3F1D
6Ø28: ED B7 CE FF B7 DØ EE 6Ø   $9ØCB
```

softkey for...

Speed

by Tim Beckmann

Speed Reader II
Davidson & Associates, Inc.
6Ø69 Groveoak Place #12
Rancho Palos Verdes, CA 90274

Requirements:
48K Apple II and up
Super IOB 1.5
A blank disk

Speed Reader II is an educational program that is meant to help you increase your reading speed. The program has quite a bit of educational value because data disks are available for many different age groups, but much of this educational value is offset by the fact that Speed Reader is protected quite heavily.

The Protection

I suspect that Speed Reader was protected by the Lock-It-Up program because parameter lists say to use the parameters for Lock-It-Up. After trying these parameters several times without success, I decided to remove the protection. This procedure should be helpful for softkeying other programs protected by Lock-It-Up.

My first hint as to how I should go about softkeying this disk was the Basic prompt that appeared when the disk booted. This usually means that using the program's own RWTS and the Swap controller with Super IOB is all that is needed to remove the protection. Unfortunately, this procedure did not work for Speed Reader because when it was protected by Lock-It-Up, the RWTS was re-written.

With this avenue closed off, my next step was to get out my nibble editor and look at each track. To my dismay, I found that tracks $Ø through $F all had different address markers. This pattern is repeated on tracks $1Ø through $1F and tracks $2Ø through $22 each have different address markers. Address markers are, as you probably have learned by now, what DOS looks for to find which sectors it is reading. In addition to this, the address end markers have been made different on each sector and the data marker on each track has been changed to D5 AA B5 from the normal D5 AA AD.

With these changes identified, I made a Super IOB controller to copy the disk. As I soon found out it was not that easy. The first time I tried to copy the disk, I got a checksum error on some sectors. After changing the controller to ignore checksum errors, the disk copied without trouble. But when I checked the softkeyed disk, I found that the catalog and the rest of the disk just contained garbaged data. The only explanation that entered my mind then was that a change was made in the read translation table on the original disk. A read translation table is a table of values that is used to decode the data read in from the disk. (For more about this see *Beneath Apple DOS* by Worth and Lechner). This would explain both the checksum errors and the garbaged data. By careful examination of the original Speed Reader DOS, I located the read translation table and found two changes made to it.

After changing the controller to read with the altered read translation table, the controller copied the disk without errors. I now thought that I was finished with the softkey, but the programs still have to be altered to work correctly. All the programs have a ⌃Z as the second character in its name. Also, some of the programs check to see if the DOS in memory is still the Lock-It-Up DOS. These program lines look like this:

```
400 IF PEEK (40324) = 173 OR
   PEEK (47094) <> Ø THEN 1ØØØ
```

We want to change these lines to look like this:

```
4ØØ IF PEEK (40324) <> 173 OR
   PEEK (47094) = Ø THEN 1ØØØ
```

The line is altered like it is so that it will work with normal DOS. We cannot just delete the line because the programs are very dependent upon the amount of memory they take up. After correcting these things the softkey is finished.

Step By Step

1) INITialize a blank disk, preferably with a fast DOS such as Pronto-DOS.

INIT HELLO

2) Install the Speed Reader controller listed at the end of this article into Super IOB and run it to copy the Speed Reader II disk.

3) Change the name of the hello program to get rid of the ⌃Z in the name.

RENAME H⌃ZELLO, HELLO

4) Change the lines in the programs that check for the Lock-It-Up DOS.

LOAD HELLO
POKE 1Ø4,32
POKE 33,3Ø

(that last POKE was to change the width of the text window so Applesoft won't insert extra spaces in the LISTing of a BASIC line).

Change line 4ØØ to look like the following:

```
4ØØ IF PEEK (40324) <> 173
   OR PEEK (47094) = Ø THEN 1ØØØ
```

Do this by LISTing line 4ØØ, then using the Escape key to put you into "escape" mode. Move to the beginning of the line using I, J, K and M keys, then hit the space bar to get out of "escape" mode. Then trace over the line using the right arrow key, making the necessary changes as the cursor passes over them.

POKE 1Ø4,8
SAVE HELLO
LOAD S⌃ZPEED 1

Change line 91 to look like the following:

```
91 CH=VAL(A$): PRINT CH: POKE 8185,CH:
  IF PEEK (40324) < > 173 OR
  PEEK (47094) = Ø THEN 11Ø
```

Caution: Notice that line 91 originally is lacking a PEEK before the first address. You must insert this PEEK or else the program will reboot.

Also change the = and the < > around in line 1Ø6

Reader II

SAVE S⊟ZPEED 1
LOAD S⊟ZPEED 2

Change lines 202 and 207 as before.

SAVE S⊟ZPEED 2
LOAD S⊟ZPEED READER II EDITOR

Change line 3 as before.

SAVE S⊟ZPEED READER II EDITOR

Your copy of Speed Reader II should now work just like the original, with the exception that you can back this one up.

About The Controller

This Super IOB controller varies from other fast controllers in these ways:

Line 1020 - Allows DOS to ignore the changed address end markers and reads the first three data statements since tracks $0-$2 are not being read.

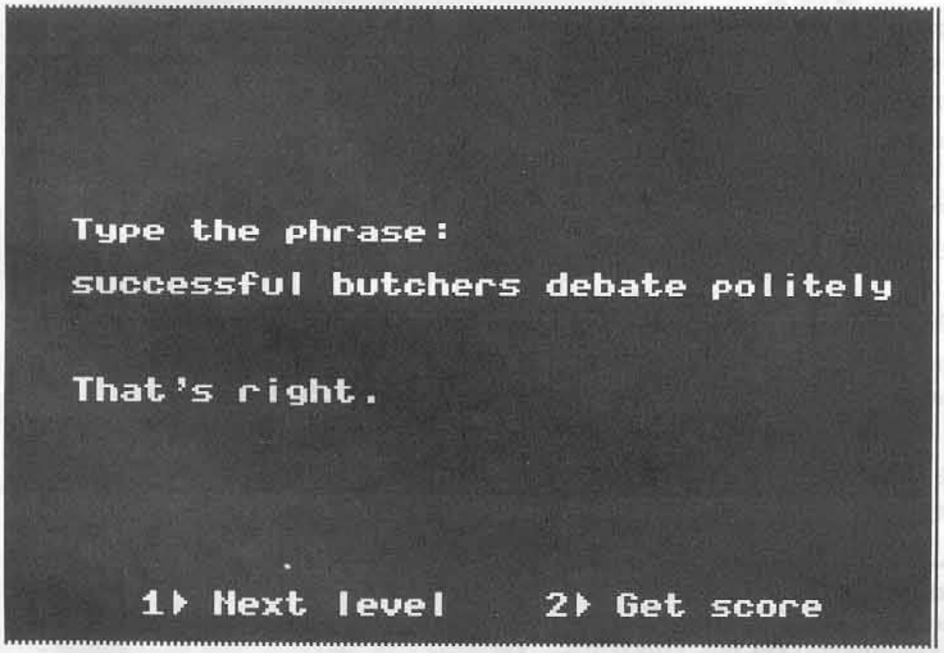
1030 - Makes the necessary changes to the read translation table and the last byte of the data marker.

1050 - Checks to see if track $10 has been reached and if it has, it restores the data so it can begin on the pattern again.

1060 - Changes the address header to the correct one for the track to be read.

1090 - Restores the correct values for the read translation table and normalizes DOS so it can write normally.

1200-1380 - Data for the address header changes.



Type the phrase:
successful butchers debate politely

That's right.

1▶ Next level   2▶ Get score

controller

```
1000 REM SPEED READER II CONTROLLER
1010 TK = 3 :LT = 4 :ST = 15 :LS = 15 :CD = WR :FAST
     = 1
1020 POKE 47426 ,24 : GOSUB 190 : GOSUB 190 :
     GOSUB 190
1030 POKE 47660 ,213 : POKE 47829 ,03 : POKE
     47356 ,181
1040 T1 = TK : GOSUB 490
1050 IF TK / 16 = INT (TK / 16 ) THEN IF TK / 16 =
     1 THEN RESTORE
1060 LT = TK + 1 : GOSUB 190 : GOSUB 610
1070 IF PEEK (BUF ) = MB THEN 1090
1080 TK = TK + 1 : IF TK < 35 THEN 1050
1090 TK = T1 : GOSUB 490 : GOSUB 230 : POKE 47660
     ,155 : POKE 47829 ,213
1100 GOSUB 610
1110 TK = PEEK (TRK ) :ST = PEEK (SCT ) : IF TK <
     35 THEN 1030
1120 HOME : PRINT "DONE^ WITH^ COPY" : END
1200 DATA 213 ,170 ,150
1210 DATA 213 ,170 ,151
1220 DATA 215 ,170 ,150
1230 DATA 215 ,170 ,151
1240 DATA 213 ,170 ,150
1250 DATA 213 ,170 ,151
1260 DATA 215 ,170 ,150
1270 DATA 215 ,170 ,151
1280 DATA 221 ,170 ,158
1290 DATA 221 ,170 ,159
1300 DATA 223 ,170 ,158
1310 DATA 223 ,170 ,159
1320 DATA 221 ,170 ,158
1330 DATA 221 ,170 ,159
1340 DATA 223 ,170 ,158
1350 DATA 223 ,170 ,159
1360 DATA 245 ,170 ,182
1370 DATA 245 ,170 ,183
1380 DATA 247 ,170 ,182
```

controller checksums

```
1000  - $356B     1230  - $A3AC
1010  - $E357     1240  - $FAB6
1020  - $5EF3     1250  - $8B4C
1030  - $9322     1260  - $6E41
1040  - $67AB     1270  - $9DA8
1050  - $8232     1280  - $B828
1060  - $BA93     1290  - $99E8
1070  - $37F5     1300  - $1425
1080  - $2435     1310  - $07CC
1090  - $158F     1320  - $FEC0
1100  - $53F1     1330  - $1314
1110  - $E0CA     1340  - $7215
1120  - $8DA8     1350  - $2DA8
1200  - $E896     1360  - $4DD4
1210  - $F590     1370  - $D192
1220  - $BC39     1380  - $FD9B
```

The Book Of Softkeys

Volumes I and II

Legends tell of the days when the ancient back issues of Hardcore COMPUTIST were readily available to anyone who wished to purchase them. Those days may be long since past, but the information contained in these ancient documents has been diligently transcribed to the pages of a modern reference work: The Book Of Softkeys.

From deep within the COMPUTIST archives comes a collection of softkeys originally contained in issues 1 through 5 and 6 through 10 (B.O.S. Volumes I and II respectively). The two volumes also contain some of the more useful programs and tutorials presented in those early issues. The books make an economical alternative to those rare (and unavailable) back issues of Hardcore COMPUTIST.

The world may end soon, so it is imperative that you obtain this valuable information before it is no longer available.

Such is the stuff that legends are made of.

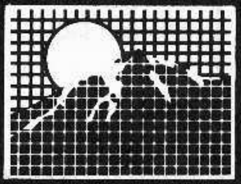
Volume I: Issues 1-5 ($12.95)

contains softkeys for: Akalabeth | Ampermagic | Apple Galaxian | Aztec | Bag of Tricks | Bill Budge's Trilogy | Buzzard Bait | Cannonball Blitz | Casino | Data Reporter | Deadline | Disk Organizer II | Egbert II Communications Disk | Hard Hat Mack | Home Accountant | Homeword | Lancaster | Magic Window II | Multi-disk Catalog | Multiplan | Pest Patrol | Prisoner II | Sammy Lightfoot | Screen Writer II | Sneakers | Spy's Demise | Starcross | Suspended | Ultima II | Visifile | Visiplot | Visitrend | Witness | Wizardry | Zork I | Zork II | Zork III | **PLUS** how-to articles and program listings of need-to-have programs used to make unprotected backups.

Volume II: Issues 6-10 ($17.95)

contains softkeys for: Apple Cider Spider | Apple Logo | Arcade Machine | The Artist | Bank Street Writer | Cannonball Blitz | Canyon Climber | Caverns of Freitag | Crush, Crumble & Chomp | Data Factory 5.Ø | DB Master | The Dic*tion*ary | Essential Data Duplicator I & III | Gold Rush | Krell Logo | Legacy of Llylgamyn | Mask Of The Sun | Minit Man | Mouskattack | Music Construction Set | Oil's Well | Pandora's Box | Robotron | Sammy Lightfoot | Screenwriter II v2.2 | Sensible Speller 4.Ø, 4.Øc, 4.1c | the Spy Strikes Back | Time Zone v1.1 | Visible Computer: 65Ø2 | Visidex | Visiterm | Zaxxon | Hayden Software | Sierra Online Software | **PLUS** the complete listing of the ultimate cracking program...Super IOB 1.5 | **and more!**

To Order: Send $17.95 + Shipping and Handling for Volume II and/or $12.95 + S&H for Volume I. Shipping and handling is $2.00 per book for US orders, $5.00 per book for foreign orders. U.S. funds drawn on U.S. banks only. Washington State orders add 7.8% sales tax. Send your orders to: **SoftKey Publishing, PO Box 110937-BK, Tacoma, WA 98411**



Are you missing a piece of the picture?

Issue	Mag $4.75	Disk $9.95	Both $12.95
40	□	□	□
39	□	□	□
38	□	□	□
37	□	□	□
36	□	□	□
35	□	□	□
34	□	□	□
33	□	□	□
32	□	□	□
31	□	□	□
30	□	□	□
29	□	□	□
28	□	□	□
27	□	□	□
26	□	□	□
25	□	□	□
24	□	□	□
23	●	□	●
22	□	□	□
21	●	□	●
20	□	□	□
19	□	□	□
18	●	□	●
☆17	□	□	□
16	□	□	□
★Book of Softkeys Vol 3			
15	●	□	●
14	●	□	●
13	●	□	●
12	●	□	●
11	●	□	●
★Book of Softkeys Vol 2 $19.95	□		
10	●	□	●
9	●	□	●
8	●	□	●
☆7	□		□
6	●	□	●
★Book of Softkeys Vol 1 $14.95	□		
4	□	□	□
3	●		●
Core 2	□		□
2	●	□	●
1	□		□
Core 1	□		□
Core 3	□	□	□
Computing 3	□	●	●
Best of Hardcore Computing	●	□	●
★Core Special $10.00 (All three CORE magazines)	□		

Special "Both" disk & magazine combination orders apply to one issue and its corresponding disk. Prices shown are for U.S., Canada, and Mexico only.

Some disks apply to more than one issue and are shown as taller boxes.

★ Book Of Softkeys volumes do not come with disks.
☆ We have a limited supply of these issues.
● Back issue is no longer available

COMPUTIST back issues and library disks are frequently referenced in current issues.

Some back issues are no longer available, but library disks can still be purchased for all back issues.

What is a library disk?

Library Disks are perfect companions for

COMPUTIST

So, if you're-

- Tired of entering lengthy Super IOB Controllers, only to find that you must have made a typo.
- Tired of entering LOOOOONG program listings.
- Tired of spending TOO MUCH TIME looking for errors.

then you need the library disks.

Each library disk contains information that would normally have to be entered by the user. Documentation for each Library Disk can be found in the corresponding issue.

Rates For Foreign Orders

- Canada and Mexico rates are identical to U.S. First Class unless otherwise specified.
- Other Foreign Back Issue Rates: $12.25 each. (includes shipping)
- Other Foreign Library Disk rates: $11.94 each. (includes shipping). Special "Both" disk and magazine combinations shown do NOT apply to Foreign orders.
- Other Foreign Book of Softkey Rates: Volume 1-$17.95 Volume 2-$22.95 (includes shipping)

Send me the back issues and/or library disks indicated:

Name____________________ ID#________

Address ____________________

City __________ State ______ Zip ______

Country __________ Phone ________

VISA _____-_____-_____-_____ Exp.______

CP39

Signature ____________________

Send check or money order to: COMPUTIST PO Box 110846-T Tacoma, WA 98411. Most orders shipped UPS so please use street address. Offer good while supply lasts. In Washington state, add 7.8% sales tax. **U.S. funds drawn on U.S. bank.**

Description of Available Back Issues.

38 *Softkeys* | Cyclod | Alternate Realty | Boulder Dash I & II | Hard Hat Mack (Revisited) | The Other Side | *Readers' Softkeys* | F-15 Strike Eagle | Championship Lode Runner | Gato V 1.3 | I, Damiano | Wilderness | Golf's Best | *Features* | The Enhanced/Unenhanced //e | Looking into Flight Simulator's DOS | *Core* | Appavarex | Installing a RAM disk into DOS 3.3 | ....

37 *Softkeys* | Under Fire | Pegasus][| Take 1 (revisited) | Flight Simulator II v1.05 (part 2) | *Readers' Softkeys* | Magic Slate | Alter Ego | Rendezvous | Quicken | Story Tree | Assembly Language Tutor | Avalon Hill games | Dark Crystal | *Features* | Playing Karateka on a //c | Track Finder | Sylk to Dif | *Core* | Breaking In: tips for beginners | Copy][Plus 6.0: a review | The DOS Alterer | ....

36 *Softkeys* | Flight Simulator II v 1.05 | AutoDuel | *Readers' Softkeys* | Critical Reading | Troll's Tale | Robot War | General Manager | Plasmania | Telarium Software | Kidwriter v1.0 | Color Me | *Features* | ScreenWriter meets Flashcard | The Bus Monitor | Mousepaint for non-Apples | *Core* | The Bard's Dressing Room | *Advanced Playing Techniques* | Championship Lode Runner | ....

35 *Softkeys* | Hi-res Cribbage | Olympic Decathlon | Revisiting F-15 Strike Eagle | Masquerade | The Hobbit | *Readers' Softkeys* | Pooyan | The Perfect Score | Alice in Wonderland | The Money Manager | Good Thinking | Rescue Raiders | *Feature* | Putting a New F8 on Your Language Card | *Core* | Exploring ProDOS by installing a CPS Clock Driver | ....

34 *Softkeys* | Crisis Mountain | Terripin Logo | Apple Logo II | Fishies 1.0 | SpellWorks | Gumball | *Readers' Softkeys* | Rescue at Rigel | Crazey Mazey | Conan | Perry Mason: The Case of the Mandarin Murder | Koronis Rift | *Feature* | More ROM Running | *Core* | Infocom Revealed | ....

33 *Softkeys* | Word Juggler | Tink! Tonk! | Sundog v2.0 | G.I. Joe & Lucas Film's Eidolon | Summer Games II | Thief | Instant Pascal | World's Greatest Football Game | *Readers' Softkeys* | Graphic Adventure #1 | Sensible Grammar & Extended Bookends | Chipwits | Hardball | King's Quest II | The World's Greatest Baseball Game | *Feature* | How to be the Sound Master | *Core* | The Mapping of Ultima IV |

32 *Softkeys* | Revisiting Music Construction Set | Cubit | Baudville Software | Hartley Software | Bridge | Early Games for Young Children | Tawala's Last Redoubt | *Readers' Softkeys* | Print Shop Companion | Kracking Vol II | Moebius | Mouse Budget, Mouse Word & Mouse Desk | Adventure Construction Set | *Feature* | Using Data Disks With Microzines | *Core* | Super IOB v1.5 a Reprint | .....

31 *Softkeys* | Trivia Fever | The Original Boston Computer Diet | Lifesaver | Synergistic Software | Blazing Paddles | Zardax | *Readers' Softkeys* | Time Zone | Tycoon | Earthly Delights | Jingle Disk | Crystal Caverns | Karate Champ | *Feature* | A Little Help With The Bard's Tale | *Core* | Black Box | Unrestricted Ampersand | ....

1 **30** *Softkeys* | Millionaire | SSI's RDOS | Fantavision | Spy vs. Spy | Dragonworld | *Readers' Softkeys* | King's Quest | Mastering the SAT | Easy as ABC | Space Shuttle | The Factory | Visidex 1.1E | Sherlock Holmes | The Bards Tale | *Feature* | Increasing Your Disk Capacity | *Core* | Ultimaker IV, an Ultima IV Character Editor | ....

29 *Softkeys* | Threshold | Checkers v2.1 | Microtype | Gen. & Organic Chemistry Series | Uptown Trivia | Murder by the Dozen | *Readers' Softkeys* | Windham's Classics | Batter Up | Evelyn Wood's Dynamic Reader | Jenny of the Prairie | Learn About Sounds in Reading | Winter Games | *Feature* | Customizing the Monitor by Adding 65C02 Disassembly | *Core* | The Animator | ....

2 **28** *Softkeys* | Ultima IV | Robot Odyssey | Rendezvous | Word Attack & Classmate | Three from Mindscape | Alphabetic Keyboarding | Hacker | Disk Director | Lode Runner | MIDI/4 | *Readers' Softkeys* | Algebra Series | Time is Money | Pitstop II | Apventure to Atlantis | *Feature* | Capturing the Hidden Archon Editor | *Core* | Fingerprint Plus: A Review | Beneath Beyond Castle Wolfenstein (part 2) | ....

27 *Softkeys* | Microzines 1-5 | Microzines 7-9 | Microzines (alternate method) | Phi Beta Filer | Sword of Kadash | *Readers' Softkeys* | Another Miner 2049er | Learning With Fuzzywomp | Bookends | Apple Logo II | Murder on the Zinderneuf | *Features* | Daleks: Exploring Artificial Intelligence | Making 32K or 16K Slave Disks | *Core* | The Games of 1985: part II | ....

26 *Softkeys* | Cannonball Blitz | Instant Recall | Gessler Spanish Software | More Stickybears | *Readers' Softkeys* | Financial Cookbook | Super Zaxxon | Wizardry | Preschool Fun | Holy Grail | Inca | 128K Zaxxon | *Feature* | ProEdit | *Core* | Games of 1985 part I | ....

25 *Softkeys* | DB Master 4.2 | Business Writer | Barron's Computer SAT | Take 1 | Bank Street Speller | Where In The World Is Carmen Sandiego | Bank Street Writer 128K | Word Challenge | *Readers' Softkeys* | Spy's Demise | Mind Prober | BC's Quest For Tires | Early Games | Homeword Speller | *Feature* | Adding IF THEN ELSE To Applesoft | *Core* | DOS To ProDOS And Back | ....

24 *Softkeys* | Electronic Arts software | Grolier software | Xyphus | F-15 Strike Eagle | Injured Engine | *Readers' Softkeys* | Mr. Robot And His Robot Factory | Applecillin II | Alphabet Zoo | Fathoms 40 | Story Maker | Early Games Matchmaker | Robots Of Dawn | *Feature* | Essential Data Duplicator copy parms | *Core* | Direct Sector Access From DOS | ..

22 *Softkeys* | Miner 2049er | Lode Runner | A2-PB1 Pinball | *Readers' Softkeys* | The Heist | Old Ironsides | Grandma's House | In Search of the Most Amazing Thing | Morloc's Tower | Marauder | Sargon III | *Features* | Customized Drive Speed Control | Super IOB version 1.5 | *Core* | The Macro System | ....

20 *Softkeys* | Sargon III | Wizardry: Proving Grounds of the Mad Overlord and Knight of Diamonds | *Reader' Softkeys* | The Report Card V1.1 | Kidwriter | *Feature* | Apple][Boot ROM Disassembly | *Core* | The Graphic Grabber v3.0 | Copy II+ 5.0: A Review | The Know-Drive: A Hardware Evaluation | An Improved BASIC/Binary Combo | ....

19 *Readers' Softkeys* | Rendezvous With Rama | Peachtree's Back To Basics Accounting System | HSD Statistics Series | Arithmetickle | Arithmekicks and Early Games for Children | *Features* | Double Your ROM Space | Towards a Better F8 ROM | The Nibbler: A Utility Program to Examine Raw Nibbles From Disk | *Core* | The Games of 1984: In Review-part II | .....

17 *Softkeys* | The Print Shop | Crossword Magic | The Standing Stones | Beer Run | Skyfox | Random House Disks | *Features* | A Tutorial For Disk Inspection and the Use Of Super IOB | S-C Macro Assembler Directives (reprint) | *Core* | The Graphic Grabber For The Print Shop | The Lone Catalog Arranger v1.0 Part 2 | ....

16 *Softkey* | Sensible Speller for ProDOS | Sideways | *Readers' Softkeys* | Rescue Raiders | Sheila | Basic Building Blocks | Artsci Programs | Crossfire | *Feature* | Secret Weapon: RAMcard | *Core* | The Controller Writer | A Fix For The Beyond Castle Wolfenstein Softkey | The Lone Catalog Arranger Part 1 | ....

7 *Softkeys* | Zaxxon | Mask of the Sun | Crush Crumble & Chomp | Snake Byte | DB Master | Mouskattack | *Features* | Making Liberated Backups That Retain Their Copy Protection | S-C Assembler: Review | Disk Directory Designer | *Core* | Corefiler: Part 1 | Upper & Lower Case Output for Zork | ....

4 Ultima II Character Editor | *Softkeys* | Ultima II | Witness | Prisoner II | Pest Patrol | Adventure Tips for Ultima II & III | Copy II Plus PARMS Update..

1 *Softkeys* | Data Reporter | Multiplan | Zork | *Features* | PARMS for Copy II Plus | No More Bugs | APT's for Choplifter & Cannonball Blitz | *'Copycard' Reviews* | Replay | Crackshot | Snapshot | Wildcard |

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 | Dragon Dungeon | ....

CORE 2 ....Utilites:

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 | ....

Hardcore Computing 3 .....

HyperDOS Creator | Menu Hello | Zyphyr Wars | Vector Graphics | Review of Bit Copiers | Boot Code Tracing | Softkey IOB | Interview with 'Mike' Markkula | ....

The Super IOB Collection

- What could possibly be better than receiving COMPUTIST every month, typing in the Super IOB controllers and deprotecting your favorite software? How about having **all the controllers ever printed in COMPUTIST** at your fingertips? With *The Super IOB Collection* Volumes I & II, you have just that and more.

- Each volume (supplied on a DOS 3.3 disk) contains at least 60 Super IOB controllers including the standard, swap, newswap and fast controllers. In addition, each disk has the Csaver program from COMPUTIST No. 13. But wait! You also get version 1.5 of Super IOB and a menu hello program that lists the available controllers and, when you select one, automatically installs it into Super IOB and RUNs the resulting program.*

- Several of the controllers deprotect the software completely with no further steps. This means that some programs are only minutes away from deprotection (with virtually no typing).

- The issue of COMPUTIST in which each controller appeared is indicated in case further steps are required to deprotect a particular program.†

Volume 1

Volume 1 of the Super IOB collection covers all the controllers appearing in COMPUTIST No. 9 through No. 26. In addition, the newswap and fast controllers from COMPUTIST No. 32 are included. The following 60 controllers are on volume 1:

Advanced Blackjack, Alphabet Zoo, Arcade Machine, Archon II, Archon, Artsci Software, Bank Street Writer, Barrons SAT, Beyond Castle Wolfenstein, BSW //c Loader, Castle Wolfenstein, Computer Preparation: SAT, Dazzle Draw, DB Master 4 Plus, Death in the Carribean, Dino Eggs, DLM Software, Electronic Arts, F-15 Strike Eagle, Fast Controller, Fathoms 40, Financial Cookbook, Gessler Software, Grandma's House, The Heist, In Search of the Most Amazing Thing, Instant Recall, Kidwriter, Lions Share, Lode Runner, Mastertype, Match Maker, Miner 2049er, Minit Man, Mufplot, Newsroom, Newswap controller, Penguin Software, Print Shop Graphic Library, Print Shop, Rendezvous with Rama, Rockys' Boots, Sargon III, Sea Dragon, Shiela, Skyfox, Snooper Troops, Standard controller, Stoneware Software, Summer Games, Super Controller, Super Zaxxon, Swap Controller, TAC, Ultima III, Word Challenge, Xyphus, Zaxxon

Volume 2

Volume 2 of the Super IOB collection covers all the controllers appearing in COMPUTIST No. 27 through No. 38. The following 65 controllers are on volume 2:

Alice in Wonderland, Alphabetic Keyboarding, Alternate Reality, Autoduel, Checkers, Chipwits, Color Me, Conan.data, Conan.prog, CopyDOS, Crisis Mountain, Disk Director, Dragonworld, Early Games, Easy as ABC, F-15 Strike Eagle, Fantavision, Fast controller, Fishies, Flight Simulator, Halley Project, Hartley Software (a), Hartley Software (b), Jenny of the Prarie, Jingle Disk, Kidwriter, Kracking Vol II, Lode Runner, LOGO II (a), LOGO II (b), Masquerade, Mastering the SAT, Microtype: The Wonderful World of Paws, Microzines 1, Microzines 2-5, Miner 2049er, Mist & View to a Kill, Murder on the Zinderneuf, Music Construction Set, Newswap controller, Olympic Decathlon, Other Side, Phi Beta Filer, Pitstop II, Print Shop Companion, RDOS, Robot War, Spy vs Spy, Standard controller, Sundog V2, Swap controller, Sword of Kadash, Synergistic Software, Tawala's last Redoubt, Terripin Logo, Threshold, Time is Money, Time Zone, Tink! Tonk!, Troll's Tale, Ultima IV, Wilderness, Word Attack & Classmate, World's Greatest Baseball, World's Greatest Football

To Order: Send $9.95 for each volume or $19.95 for a complete package that includes: both disks, a reprint of "Disk Inspection and the use of Super IOB" and COMPUTIST No. 32. U.S. funds drawn on U.S. banks. Foreign orders (other than Canada or Mexico) add 20% shipping. Washington state residents add 7.8% sales tax. Mail orders to: Super IOB Collection; POB 110846; Tacoma, WA 98411

*Requires at least 64K of memory.

†Although some controllers will completely deprotect the program they were designed for, some will not and therefore require their corresponding issue of COMPUTIST to complete the deprotection procedure.