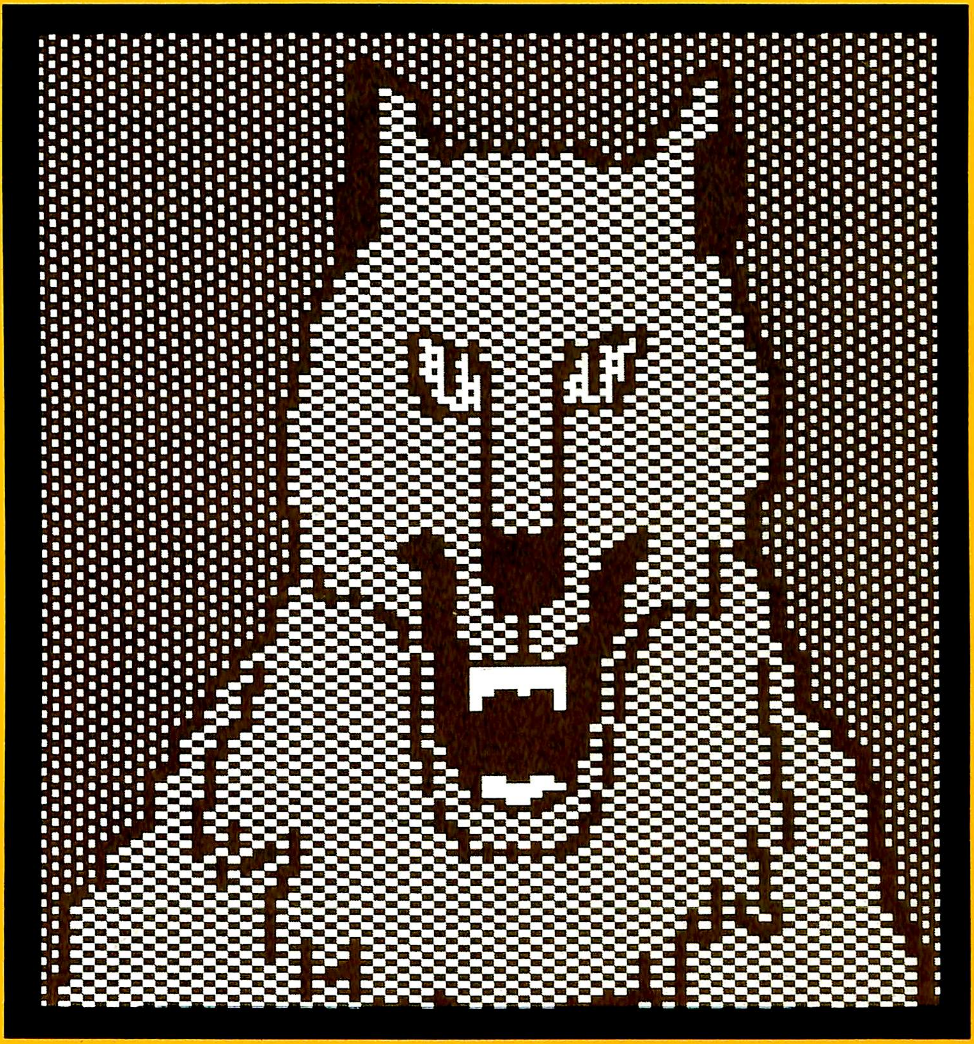
For The Serious User Of Apple][Computers

COMPUTIST

Issue No. 28 $3.75


(Page 9)

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

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

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

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

What Is A Softkey Anyway? 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 from normal text by being indented and bold. An example is:

```
PR#6
```

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

Control characters are indicated by being boxed. An example is:

6$\boxed{\text{P}}$

To complete this command, you must first type the number 6 and then place one finger on the CTRL key and one finger on the P key.

Requirements: Most of the programs and softkeys which appear in COMPUTIST require one of the Apple][series of computers and at least on disk drive with DOS 3.3. Occasionally, some programs and procedures have special requirements. The prerequisites for deprotection techniques or programs will always be listed at the beginning of the article under the "Requirements:" heading.

Software Recommendations: The following programs (or similar ones) are strongly recommended for readers who wish to obtain the most benefit from our articles:

1) **Applesoft Program Editor** such as Global Program Line Editor (GPLE).
2) **Sector Editor** such as DiskEdit, ZAP from Bag of Tricks or Tricky Dick from The CIA.
3) **Disk Search Utility** such as The Inspector, The Tracer from The CIA or The CORE Disk Searcher.
4) **Assembler** such as the S-C Assembler or Merlin/Big Mac.
5) **Bit Copy Program** such as Copy][Plus, Locksmith or The Essential Data Duplicator
6) **Text Editor** capable of producing normal sequential text files such as Applewriter][, Magic Window][or Screenwriter][.

You will also find COPYA, FID and MUFFIN from the DOS 3.3 System Master Disk useful.

Super IOB: This program has most recently appeared in COMPUTIST No. 22. Several softkey procedures will make use of a Super IOB controller, a small program that must be keyed into the middle of Super IOB. The controller changes Super IOB so that it can copy different disks. To get the latest version of this program, you may order COMPUTIST No. 22 as a back issue or order Program Library Disk No. 22.

RESET Into The Monitor: Some softkey procedures require that the user be able to enter the Apple's system monitor during the execution of a copy protected program. 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 of COMPUTIST No. 6 or the "Dual ROM's" article in COMPUTIST No. 19.

Apple //e - Apple //c: Install a modified CD ROM on the computer's motherboard. Clay Harrell's company (Cutting Edge Ent.; Box 43234 Ren Cen Station-HC; Detroit, MI 48243) sells a hardware device that will give you this ability. Making this modification to an Apple //c will void its warranty but the increased ability to remove copy protection may justify it.

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 Peter Leichner, Quality Software, $19.95; *Assembly Language For The Applesoft Programmer*, Roy Meyers and C.W. Finley, Addison Wesley, $16.95; and *What's Where In The Apple*, William Lubert, Micro Ink., $24.95.

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. To understand this format, you must first understand the formatted LIST feature of Applesoft.

An illustration- If you strike these keys:

```
10 HOME:REMCLEAR SCREEN
```

a program will be stored in the computer's memory. Strangely, this program will *not* have a LIST that is exactly as you typed it. Instead, the LIST will look like this:

```
10 HOME : REM CLEAR SCREEN
```

Programs don't usually LIST the same as they were keyed in 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 command words. The space inserted after these command words can be misleading. For example, if you want a program to have a list like this:

```
10 DATA 67,45,54,52
```

you would have to omit the space directly after the DATA command word. If you were to key in the space directly after the DATA command word, the LIST of the program would look like this:

```
10 DATA  67,45,54,52
```

This LIST is different from the LIST you wanted. The number of spaces you key after DATA and REM command words is very important.

All of this brings us to the COMPUTIST LISTing format. In a BASIC LISTing, there are two types of spaces; spaces that don't matter whether they are keyed or not and spaces that must be keyed. Spaces that must be keyed in are printed as delta characters (▵). All other spaces in a COMPUTIST BASIC listing are put there for easier reading and it doesn't matter whether you type them or not.

There is one exception: If you want your checksums (See "Computing Checksums" section) to match up, you *must not* key in any spaces after a DATA command word unless they are marked by delta characters.

Keying In Hexdumps: Machine language programs are printed in COMPUTIST as both source code and hexdumps. Only one of these formats need be keyed in to get a machine language program. Hexdumps are the shortest and easiest format to type in.

To key in hexdumps, you must first enter the monitor:

```
CALL -151
```

Now 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). If you hear a beep, you will know that you have typed something incorrectly and must retype that line.

When finished, return to BASIC with a:

```
E003G
```

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

Keying In Source Code The source code portion of a machine language program is provided only to better explain the program's operation. If you wish to key it in, you will need an assembler. The S-C Assembler is used to generate all source code printed in COMPUTIST. 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 just for this purpose was printed in COMPUTIST No. 17. To translate source code, you will need to understand the directives of your assembler and convert the directives used in the source code listing to similar directives used by your assembler.

Computing Checksums Checksums are four digit hexadecimal numbers which verify whether or not you keyed a program exactly as it was printed 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 programs appeared in COMPUTIST No. 1 and The Best of Hardcore Computing. An update to CHECKSOFT appeared in COMPUTIST No. 18. If the checksums these programs create on your computer match the checksums accompanying the program in the magazine, then you keyed in the program correctly. If not, the program is incorrect at the line where the first checksum differs.

1) To compute CHECKSOFT checksums:

```
LOAD filename
BRUNCHECKSOFT
```

Get the checksums with

```
&
```

And correct the program where the checksums differ.

2) To compute CHECKBIN checksums:

```
CALL -151
BLOAD 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 $\boxed{\text{Y}}$.

xxx.xxx$\boxed{\text{Y}}$

And correct the lines at which the checksums differ.

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.

Our editorial policy is that we do NOT condone software piracy, but we do believe that honest 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 application programs to meet his or her needs.

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.

renew your freedom

Check your mailing label to see if you need to renew your subscription. And if you think you might forget when that fatal time arrives, renew right now. Just use the order blank below.

if you're moving...

Let us know right away or at least 30 days in advance so that you won't miss a single issue. Just write your new address here, and paste your present address label in the order form below and send it to us. Issues missed due to non-receipt of this Change-of-Address may be aquired at the regular back-issue rates. **My new address is:**

Name ____________________
Address ____________________
City __________ State ______ Zip ______
Country __________ Phone __________

We are NOT PIRATES

but we're not fools, either.

We're serious programmers and software users who just want to have backup copies of any software we own. COMPUTIST magazine shows us HOW TO MAKE BACKUPS OF COMMERCIAL SOFTWARE regardless of the maker's attempt to stop us from having legal copies. Don't let them stop you from protecting your own rights.
Remove copy-protection from your valuable library of expensive software. The publisher of COMPUTIST has been showing subscribers how to unlock and modify commercial software for the past 4 years. Don't be one of the users abused by user-FIENDLY locked-up software. Subscribe.

6 issue SUBSCRIPTION RATES:
U.S.: $20 first class: $24
Canada, Mexico: $34
Other Foreign: $60
SAMPLE COPY:
U.S.: $4.75 Foreign: $5.70
US funds drawn on U.S. bank.
In Washington add 7.8% tax.
Send check or money order to:
COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

☐ NEW subscriber ☐ Renew my subscription
Name ____________________ ID# ______
Address ____________________
City __________ State ______ Zip ______
Country __________ Phone __________
______-______-______-______ Exp. ______
Signature ____________________ CP28

REWARD
up to
$100.00

Flight Simulator II

Sub Logic

If your softkey to this program is published in COMPUTIST.

REWARD
up to
$90.00

Gutenburg Jr. & Sr.

Micromation LTD

If your softkey to this program is published in COMPUTIST.

REWARD
up to
$80.00

The Handlers

Silicon Valley Systems

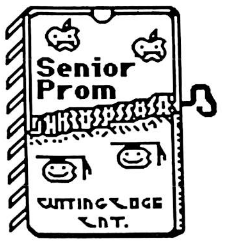
If your softkey to this program is published in COMPUTIST.

Unlock the Software Secrets

The Senior PROM //c, //e: A unique device which combines the features of a copy card, sector editor, memory detective, nibble copier, absolute reset, and much more into a single affordable tool. Provides Assembly Language utilities such as an Assembler, Step and Trace, and more. ROM resident operating system with disk copy, format, edit, and protected disk utilities, all without booting DOS first!

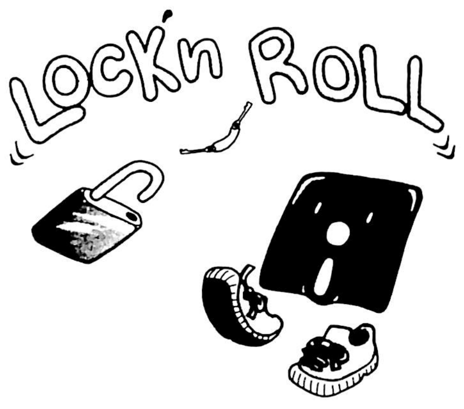
All the **Senior PROM**'s features are in **ROM**, instantly available when needed. Undetectable by any software or hardware, does not use a peripheral slot. Extensive documentation & guide to copy-protection. **Nothing else like it available for the //c!**

Economically priced at **$79.95** for prepaid orders with a check or money order. Credit Card orders available for **$88.95**. Please specify **//c**, or **//e** Standard or Enhanced ROMs.



Cutting Edge Enterprises
Box 43234 Ren Cen Station
Detroit, MI 48243
313-349-2954 Modem.
For **VISA** **orders call**
317-743-4041, 10-5 E.S.T.
not intended for illegal use.

THE MOST POWERFUL BACK-UP UTILITY
YOU'VE EVER SEEN...



***$59.95** add $5 ship.*

* Back-ups 1/2, 1/4, 3/4 tracks.
* Automatic back-up options.
* No parms needed for most of the back-ups.
* Excellent DOS copy on flip side.

Ask for our other products RAM-LOCK(for L-Smith)
And also SHOUGI(Japanese chess-type game.)

ART GALLERY
Yoshinoya Bldg,438 Sasu-machi,Chofu-shi
Tokyo 182,Japan
Send money or check. VISA/MASTER CARD accepted.

BULLSHIRT!

by bobby
PHD, phD, Phd, PhD, BHT, BLT, MSG, McDLT

Requirements
some $$$
postage stamp
Envelope
Pen or compatible word processor
scissors (optional)

Now that HARDCORE has changed its name to plain old 'computist', I've been assigned the regrettably redundant responsibility of changing all the Diskbusters T-shirts. Instead, using my own imaginary initiative, I've come up with a reasonable facsimile of an original idea...

"Why not," I asked myself, "call all the old Diskbusters T-shirt something catchy like: Hardcore Classic?"

"Wow!" I answered myself, "Then we could sell people the NEW computist T-shirts this summer!"

(I've heard that some people suspect that there's a rumor that such an artifact does not yet exist...VAPORWEAR...)"

If you like my idea, then PLEASE order the CLASSIC Hardcore Computist DiskBusters T-shirt (simply known as CLASSIC T-shirt)

1) Fill out the order form below using the Pen and then insert the order form with appropriate $$$ or acceptable substitutes into the envelope, and then pen-in the address onto the envelope, and then attach the stamp and that's it.

Enjoy!

I want the CLASSIC! so rush me _______ (total) Classic DISKBUSTERS T-SHIRTS in the sizes indicated below. I have enclosed $9.95 (plus tax and shipping) for each shirt.

ADULT MENS: _____Small _____Medium _____Large _____Xtra large

Name ______________________________ ID# __________

Address ______________________________

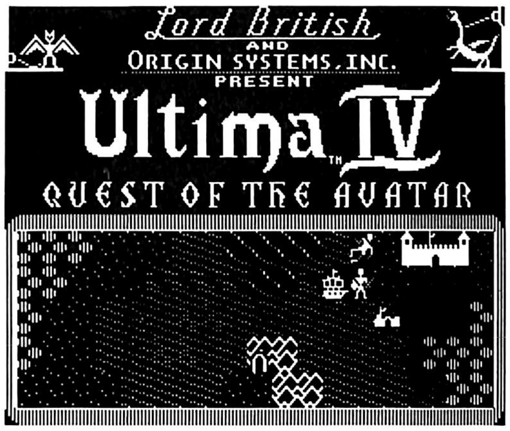
City ______________ State ________ Zip ____________

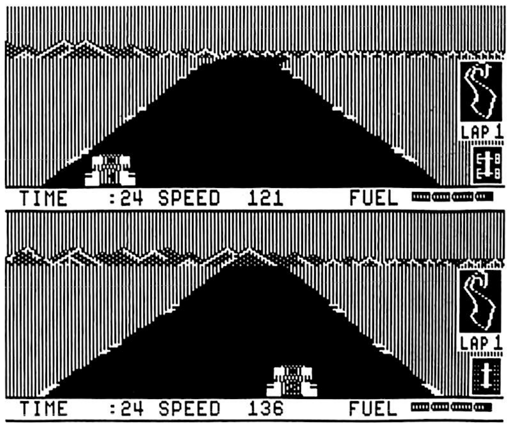
Country ______________ Phone ______________

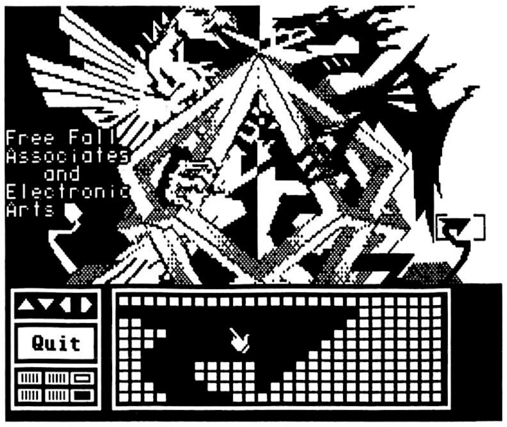
VISA MC ______-______-______-______ Exp. ________

Signature ______________________________ CP28

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







This month's cover:
Graphics from Ultrasoft's "Serpent's Star."

SUBSCRIPTIONS: Rates (for 6 issues): U.S. $20, U.S. 1st Class $24, Canada & Mexico $34, Foreign $60. Direct inquiries to: **COMPUTIST, Subscription Department, PO Box 110846-T, Tacoma, WA 98411.** Please include address label with correspondence.
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 28

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

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.

Zapped

I have a question. A friend of mine had Artworx Strip Poker in the disk drive and was trying to figure out how to deprotect it, when his computer line was struck by lightning. This caused his language card to freak out and by some weird instance, it deprotected Strip Poker! How!?!

Baron Von Zonker

Mr. Zonker: As I see it, there are three possible reasons for this event:

1*) The lightning caused the computer to execute a routine which was responsible for deprotecting the game.*

2*) The psychic energy caused by your friend's intense desire to deprotect the game was amplified by the lightning thus producing a deprotected copy.*

3*) Perhaps God is on our side of the copy protection rat race and this was his way of getting back at the system.*

The EDD Syndrome

I would like to tell you how much I enjoy your magazine. I think it has helped me to understand the Apple][computer operations more than any other single book or magazine. It does take a lot of study to understand programming and protection techniques however, and I am still learning. I am writing to tell you of some of my experiences using your magazine articles to softkey one of my programs.

The program that I started on was EDD III. I used the article on EDD I in COMPUTIST No. 8, pg 24 and the material in COMPUTIST No. 1Ø, pg 7 to get started. I booted up my copy of EDD III and went into the monitor using the modified EPROM technique described in the "Double Your ROM Space" article by Ray Darrah in COMPUTIST No. 19, pg 9. (NOTE: I used a 2764 EPROM technique which I am going to write up for a possible future article.)

I followed the steps described for EDD I with the changes suggested for EDD III. My copy is dated March 1, 1985 so I changed the address at $4026 to 53. After following these steps I saved the program and tried it. No luck! It would not boot, so I decided to see what I could do.

I rebooted EDD III and reset into the monitor when the program stops at the title screen. I looked at the memory area at $6ØØØ and there was already code there. Moving the $8ØØ page up to this address had overwritten something. This code extended from $6ØØØ to $62FF so I used a 63ØØ<Ø8ØØ.Ø8FFM command to move page 8 to the open $63ØØ area. I then booted a slave disk as described in the basic COMPUTIST article, moved the $8ØØ page back, and continued on. I should mention at this point that I inserted a $6Ø at $3EE8 and $3EE9 since the deleted operation occupied both positions. I don't know if this makes a difference here (I am still studying assembly language) but it keeps all of the other code at the same address. I tried saving this and booting it, but no luck. I kept trying things and I could get my copies to start, but they would stop unexpectedly. Finally, I was desperate. I had a version that would work if I called the program start address. I thought maybe the DOS pointers were not being restored properly so I decided to do something I had never heard done before! I did a massive BSAVE from $ØØØØ to $63ØØ! This worked (surprised me) after a fashion but I was not satisfied. It loaded garbage on the screen and required a "CALL 2Ø45" to start. I continued to examine the program code to find a better way to do it. I rebooted the master copy of EDD III and took a look at the new JSR at $Ø7FD and there it was! It had a JSR to the code at $6ØØØ already there, which was getting overwritten by the JSR to $4ØØØ. I looked ahead of this and it was clear so I added the JSR $4ØØØ at $Ø7FA, BSAVED it all to $63ØØ and it worked! I had all of EDD III except the hi-res introductory picture in COPYA format. I modified the Hello program to BRUN EDD III and the final copy boots in about 25 seconds. The complete procedure is given below.

Summary

1) RUN EDD III.

2) Reset into the monitor.

3) Change $3EE8 to $6Ø.

3EE8:6Ø

4) Change $3EE9 to $6Ø. (Replaces 2 byte command)

3EE9:6Ø

5) Move memory

63ØØ<Ø8ØØ.Ø8FFM

6) Boot initialized slave disk

C6ØØG(for slot 6)

7) Move memory back

Ø8ØØ<63ØØ.63FFM

8) Insert the COMPUTIST code at $4ØØØ

```
4ØØØ:A9  ØØ  8D  F2  Ø3  A9  C6  8D
4ØØ8:F3  Ø3  2Ø  6F  FB  A9  FØ  85
4Ø1Ø:36  A9  FD  85  37  A9  1B  85
4Ø18:38  A9  FD  85  39  A9  6Ø  85
4Ø2Ø:ØA  A9  6Ø  85  ØB  4C  5E  Ø9
```

9) Change start address

4Ø26:53 Ø9

10) Insert the new start JSR

Ø7FA:4C ØØ 4Ø

11) Save EDD III (From $7FA TO $62FF)

BSAVE EDD III, A$Ø7FA, L$5BØ6

12) That's it. Modify HELLO to display "BRUNNING EDD III" and then BRUN it automatically if you like. It is now completely broken.

M. A. Todd
Ft. Walton Beach, FL

What about my Hard Disk?

As an avid reader of your fine publication, but somewhat of a novice at the computer; I have been successful in creating a COPYAable copy of Sensible Speller DOS 3.3 that runs just as Lamont Cranston said it should. However, I just purchased a 1Ø megabyte Sider hard disk and would very much like to copy Sensible Speller onto this drive for obvious reasons. My question is this, "Can you, or any of your readers assist me to accomplish this?"

I can transfer Sensible Speller to the hard disk with no problem, but when I boot it will dump me into the monitor.

Samuel Tenney
Ontario, CA

Mr. Tenney: A Softkey can generally be divided into two broad catagories:

1) A Broken file: The result is file(s) that can be BSAVEd, BLOADed, SAVEd and LOADed.

2) A COPYAable disk: The resulting disk is COPYAable and therefore requires most every track and sector of the disk, especially the boot tracks.

A program that goes through the first type of softkey can usually be put on to any hard disk without much difficulty. Usually these programs have no disk access during the program.

Unfortunately for hard disk owners, the more sophisticated programs usually fit into the second category. This is the case with Sensible Speller. These programs either have disk access during their execution or are loaded directly into memory rather than having a file structure.

Due to the nature of hard disks, it is VERY difficult (sometimes impossible) to move these COPYAable disks to hard disks. To accomplish such a thing, you would have to follow the boot sequence of the COPYAable disk (a lot of knowledge about machine language and DOS is required) to figure out what sectors are loaded where and then convert this information to something your hard disk can emulate by writing a new boot routine.

Checksum Confusion

Gentlemen:

Too many times in your BASIC listings of programs, the author will "decorate" the beginning of the listing with slashes and backslashes. What use is it to include checksums for the programs that include backslashes? I have an Apple][which, to my knowledge, will not print blackslashes. Please refrain from printing backslashes in program listings or tell me how I can enter them from an Apple][. Until I can print a backslash, the checksums are useless and I must proofread my entire listing.

P.S. Core's "Wheel of Money" on page 26 of COMPUTIST No. 23 is a listing that uses the backslash in lines 10, 20, 80, and 90.

Michael L. Knickman
Hampstead, MD

Mr. Knickman: A little known fact about our checksum program (Checksoft) is that it ignores everything on a line following a REM command. We at COMPUTIST certainly don't expect our readers to key in our fancy (yet sometimes bothersome) REMs.

If there is a line of a program containing a REM that you can't type (or don't want to), all you have to do is type the line number followed by the command word REM and nothing else. The checksums will be the same no matter what (if anything) you put after a REM.

If it Weren't for Bad Luck ...

I have been reading your magazine for about a year-and-a-half now and enjoying it thoroughly! I have deprotected many programs with your IOB. There are, however some problems I have run into along the way, and would greatly appreciate you or my fellow readers' help in solving them.

1) The latest update of IOB (1.5) does not function. It comes up on the screen and says it's reading and writing, but neither of my drives ever gets activated!!!

2) The softkey update for EDD III did not work for my version, dated May 25, 1984.

3) The softkey for F-15 did not work on my version. The tracks my copy used for protection were 09 and 0A. Any help on that one?

4) Stickybear! I've tried every method seen in COMPUTIST but still no backup. I have ABC and Numbers.

And now, if I might offer a little help. Regarding the softkey for Archon in COMPUTIST No. 21. The modifications to track 2, sector 0, were incorrect. The last two lines in the machine code should have read:

```
$F0:  48  A9  EA  8D  00  A0  8D  4D
$F8:  A1  8D  02  A0  68  4C  D4  BC
```

Last but not least, if you must print the latest copy parameters, (available by the way, to any registered owner), you could print those for Locksmith 5.0 and Copy][Plus, not just EDD!!! My secondary objection to this is the use of four pages of available space for softkeys, APT's and other great information!!!

Unless this letter gets printed, my chances for assistance are narrowed. Please, I've written before. I'm desperate!!!

Michael Ferreira
Santa Rosa, CA

Mr. Ferreira: You are wrong about Super IOB v1.5. It works perfectly as we printed it. Either you typed something wrong or forgot to key in the new hexdump.

Perhaps the softkey for EDD III in the input column of this issue will work for your version.

It has come to our attention that there are several verisons of F-15 Strike Eagle. Therefore I can only suggest that you stay tuned to COMPUTIST for further episodes in the F-15 saga. A similar comment works for Sticky Bear stuff.

Finally, Central Point Software and Alpha Logic Business Systems have requested that we not print parameters for their copy programs (COMPUTIST is a registered owner of both). In compliance with their requests, COMPUTIST hasn't printed any such parameters since issue No. 4.

On the other hand, the nice folks from Utilico Microware frequently supply us with parameters specifically for the purpose of printing them. We feel that enough of our readers are interested in this information to warrant its publication.

Black Rose Rebuttal

The letter from Black Rose of Pirates Cove is disgusting. There is no defense for software piracy. It is stealing plain and simple. Mental midgets like Black Rose attempt to justify this stealing with some twisted have-have not logic when in fact there is no justification for stealing on any grounds. The argument that multimillion dollar corporations should never miss the little bit he steals just doesn't hold water. That's no different than saying a department store won't miss the little item one shoplifts, or that cheating insurance companies is okay because they can afford it. Stealing is stealing. It doesn't matter if Black Rose breaks into his (her?) neighbor's home and takes their belongings or if Blackie takes or gives an unauthorized copy of software, it is still stealing plain and simple.

Black Rose should check his math. I don't know how accurate his information on software revenues are, but 20-50 million stolen through software piracy is about 9-20% of 250 million, not 1/500th. I wonder how Black Rose would feel if a few thousand people were stealing 9-20% of his annual pre-tax salary every year? I'll bet he would sing a different tune. I don't blame Black Rose for not signing his real name. I wouldn't either if I were a low-life criminal like him.

As long as people like Black Rose are around, those of us who are honest consumers will have to pay for his disgusting criminal acts through higher software prices. As for myself, I buy unprotected software whenever possible.

input

Beagle Bros., Apple, Roger Wagner, and some other very good companies now sell software which is unprotected. Unfortunately, there are some very good programs that are not free of copy protection, like Print Shop, Sargon III, etc. If I knew Black Rose personally, I would inform the software manufacturers whose products he was stealing and offer to testify against him in court if they wished to prosecute. Unfortunately the cost of prosecuting petty low-lifes like Black Rose makes it unlikely that he will ever be punished.

Donald G. Moses
Incline Village, NV

Milliken Softkey

I have a softkey for the Milliken Math Sequences. The program that I worked out the following softkey for is the fractions program.

Materials:
One blank diskette
COPYA from system master diskette
Sector Editor of some type
MASTER CREATE from system master diskette

Procedure:

1) Boot up the COPYA program.

RUN COPYA

2) When you are asked for the source slot press CTRL-RESET.

3) Enter the monitor and override all DOS error checking.

```
CALL-151
B942:18
```

4) Return to BASIC

CTRL C

5) Delete line 70 of COPYA

70

6) Run COPYA and follow normal procedures for making a copy.

RUN

7) When the copy is completed RUN your sector editor and change the first two bytes of track $11, sector 0, from A0 F1 to 04 11.

8) BRUN MASTER CREATE (on System Master diskette). The greeting program should be called "BOOT".

You should now have a working copy.

William S. Hughes
Indian Head, MD

APT Plus Adventure Tip

Here is what I think might be considered an APT for choplifter. It allows you to play forever, unless you get yourself killed.

Land beside some of the men, but do not let them get inside of your helicopter. Take off and land about half a screen away. Wait till men the get as near to the helicopter as possible without boarding it, and then repeat the process.

When you get to your base, the computer will not count the man as having arrived at the base. Because of this, there will be only 63 men left, but the computer thinks there are 64 left. The computer will not be able to know if all 64 have died. Because of this you can play as long as you can survive, not worrying about the men. I use this so that I can practice killing tanks and planes. Maybe someone can figure out why this happens?

Also, here are some adventure tips for perhaps the weirdest adventure game ever made, MINDWHEEL.

Don't cast the copy of the fear sonnet aside, use your sense. If you feel the game is depriving you of seeing the images of other minds, like the tyranny of the Generalissmo, or the night in the courtyard, cheer up! All the answers to the fear sonnet are in the last two sentences.

That pinecone fits perfectly into the door at the entrance into Eva Feins mind, but that's not all it takes to open the door.

Finally, some people have asked me how I get these adventure tips. It's very easy. First take a formatted disk and save your game on it. You should save it when you're on the side you wish to examine. Then take out a sector editor, and look at Track 0. All that garbage is what tells the program what was happening at the time you save the game. Try changing most of the data (PAST SECTORS 1 AND 2) to FFs. Next, try to RESUME NOVEL. If you get a BOOKMARK ABORTED message, you were to curious and messed up some important stuff. Your screen should start filling with every possible message the program could ever give you. Stop it with ESC so that you have time to read it. This should give you some info on the inner workings of the game.

I am currently woking on decoding the way this information is stored I'll write soon with more information on the subject.

Ben Yongdahl
Sioux Falls, SD

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.

Mouse Calc	Apple Computer
Apple Business Graphics	Apple Computer
Flight Simulator II	Sub Logic
Factory	Sunburst Communicating
Jane	Arktronics
Visiblend	Microlab
Lifesaver	Microlab
Catalyst	Quark, Inc.
Gutenburg Jr. & Sr.	Micromation LTD
Prime Plotter	Primesoft Corp.
Zardax	Computer Solutions
The Handlers	Silicon Valley Systems
The Apple's Core: Parts 1-3	The Professor
King's Quest	Sierra On-line
Fun Bunch	Unicorn
Willy Byte ...	Data Trek
Trivia Fever	Professional Software Inc
Terrapin Logo V2.00	Terrapin Software
The Boston Computer Diet	Scarborough Systems
Conan	Datasoft
Cycloid	Sirius Software
Crisis Mountain	Synergistic Software
Adventure	Microsoft
Agent U.S.A.	Scholastic

readers' softkey & copy exchange

Ken Black's softkey for...

Algebra Series

Edu-Ware Services, Inc.
P.O. Box 22222
Agoura, CA 91301

Requirements:
Apple][
COPYA or FID
A blank disk

As almost everyone knows, educational software gets lots of abuse from curious little fingers. The Algebra series of disks from Edu-Ware are a good teaching aid and provide enough screen graphics to keep you on course through the lessons provided. Very lucky are we, that the backups are simple and easy and can save us our valuable original. I have the entire series of these disks and have found this procedure to work on all of them. I also believe that this will work on any of Edu-Ware's other programs.

The protection on this disk is rather simple in that the disk is in a normal format except that the address and data epilogues are changed. It can be CATALOGed by defeating DOS's error checking. This means we can probably use FID or COPYA to transfer the files. It will be faster to use COPYA to copy the disk and add a proper DOS later.

The Procedure

1) Get out COPYA and RUN it.

2) When it asks for the drive numbers, break out of the program by typing
CTRL C.

3) Enter the monitor and defeat DOS's error checking.

CALL-151
B942:18

4) Return to BASIC and delete line 70 (to avoid reloading the machine language part).

CTRL C
70

5) Restart COPYA and copy the Algebra disk to a blank disk.

6) Now put a normal (or fast) DOS on the new disk using MASTER CREATE or a similar utility, following the instructions.

You now have a backup of your delicate educational disk. I highly recommend a fast DOS to speed the program along. It makes a great deal of difference. Otherwise, class may be over before the boot is finished.

Ken Black's softkey for...

Time is Money

Turning Point Software
11A Main St.
Watertown, MA 02172

Requirements:
Apple][
Super IOB 1.5
A blank disk
The following are optional:
A disk search program
A sector editor
A copy program that can skip tracks or ignore errors

Time is Money is another home accounting program that is rather simple to use and has some nice features. As with all programs of this type you must boot the original to get the program going even if you have a data disk. This could lead to premature disk failure, so I decided to make a backup to keep the original in safe condition. My attempts to make a backup didn't go too well. Even COPYA seemed to work up to a point. After doing some checking I found that the problem was with track $22. It just didn't want to copy. Some of the time the copy would boot but the message "Not Master Disk" would appear and the program would go no further. Fortunately, the disk can be softkeyed rather easily.

The Protection

This disk appears to use a normal DOS 3.3 format up to track $22. It will copy fine to that point. Track $22 contains nothing but $FF bytes and some markers. If this strange track isn't there we get the nice message about the disk. What we need to do here is make this track in a normal format like the others and find the check routine and disable it.

In track 5, sector $F, there is a routine that accesses the disk. Look for the bytes $BD 8C C0. This translates to LDA C08C,X, which reads a byte off the disk. This is unusual, because it is not a track DOS usually resides on (a clue!). Replacing the LDA ($BD) with an RTS ($60) sometimes works because the protection code may simply exit with the carry flag changed, if the desired data was not found. In this case, store a $60 at byte $19 (on your new backup) and you will have a working copy.

The Procedure

Note: the version number of this Time Is Money disk was "1.0 F".

1) Try to catalog the original Time Is Money disk and make a note of the disk volume number. Don't forget to write protect the original.

2) Put the Super IOB controller below into Super IOB 1.5.

3) Start the program and let it format the blank disk with the same volume number as the original.

4) Copy tracks 0-$21 of Time Is Money. Super IOB will do the sector edit for you.

If that doesn't work, you might want to try this:

1) Initialize a blank disk (with the original's volume number).

2) Copy the original to the backup with the exception of the strange track.

3) With a disk searcher, look for the bytes BD 8C C0. The one you want is probably not in tracks 0-2. Replace the most likely one (in this case it was at byte $19, track 5, sector $F) with a $60, or maybe a $18 60, and write it back to the copy.

You should now be the proud owner of a COPYAable, fully functional, backup that you can use without the worry of damage to the original. Now can anyone tell me why my account is so empty?

controller

```
1000 REM TIME IS MONEY CONTROLLER
1010 TK=0 :LT=34 :ST=15 :LS=15 :CD=WR :FAST
     =1
1020 GOSUB 490 : GOSUB 610 :T1=TK :TK=PEEK (TRK
     ) - 1 : RESTORE : GOSUB 310 :TK=T1
1030 GOSUB 490 : GOSUB 610 : IF PEEK (TRK ) = LT
     THEN 1050
1040 TK=PEEK (TRK ) :ST=PEEK (SCT ) : GOTO 1020
1050 HOME : PRINT "COPYDONE" : END
1100 DATA 1^ CHANGES ,5 ,15 ,25 ,96
```

controller checksums

```
1000  - $356B     1040  - $D1CC
1010  - $0715     1050  - $B5D7
1020  - $CE56     1100  - $9B50
1030  - $7CE8
```

readers' softkey & copy exchange

A.L. Head, Jr.'s softkey for...

Pitstop II

EPYX
1Ø43 Kiel Ct.
Sunnyvale, CA 94Ø89

Requirements:
48K Apple][
Super IOB v1.5
A blank disk
DOS 3.3 Master
Pitstop II

Pitstop II is a racing game with outstanding hi-res graphics. A joystick facilitates the play of the game. Examination of the disk using a nibble editor shows that Epyx has changed the address and data epilogues from $DE $AA to $FF $FF. The checksums are all standard. First, I decided to see if I could read the tracks and sectors by patching DOS for the epilogue changes that I found. This worked fine using the Tricky Dick or Copy II Plus sector editors. There appeared to be no code on tracks $Ø3 through $1Ø although someone's name, address and telephone number are written to track $Ø6, sector $Ø9. Also, track $ØA, sector $Ø5 appears to contain bytes that could be used by a nibble count routine.

Examination of track $11 shows a standard DOS 3.3 VTOC. The one and only catalog sector is located at track $11, sector $ØF in its standard position. One binary file, BOOT, is listed, a rather descriptive name. This file has a length of $FF bytes and is loaded at $4ØØØ in memory. At this point I decided to make a copy. I first used the manual sector copy from Copy II Plus and changed parameters 5C, 5D, 66 and 67 to change the wanted epilogue bytes for the address and data epilogues. This copy would not boot but would clear memory, recalibrate and try again. This indicated that there is probably a signature check or a nibble count routine contained in the initial boot stage.

On a hunch I decided to replace DOS on the disk with a standard DOS 3.3, patch it to ignore epilogue checks and see if I could BRUN BOOT. It worked like a charm. Just to see what was going on, I next installed the The CIA Tracker, patched DOS in memory and BRUN BOOT. After accessing tracks $11, $12, and $13, the hi-res display comes on, eliminating the text window that the Tracker uses. From there, I couldn't watch the tracks it accessed.

I then BLOADed BOOT into memory and examined the code beginning at $4ØØØ. The graphics mode and pages are set beginning at $4ØØD. I NOPped all of these with $EA's in memory and then gave the command $4ØØØG. The change killed the hi-res display and allowed the Tracker to display the tracks and sectors accessed. Restarting, Tracker revealed that the BOOT program loads the game directly into memory from tracks $16-$22. Further study found the entry point for the game is at $ØAØØ. Tracks $14 and $15 are empty.

The Softkey

The plan will be to initialize a blank disk with DOS 3.3 (or similar) and use Super IOB 1.5 to normalize tracks $Ø3 through $22. At this point you could simply BRUN BOOT and start the game, but that's not good enough. A better way would be to patch DOS to BRUN BOOT instead of trying to RUN it.

1) Boot up your DOS 3.3 master.

2) Fix DOS to BRUN (instead of RUN) the hello program.

9E42:34

3) Initialize the blank disk with the modified DOS and "BOOT" for the filename.

INIT BOOT

4) Install the controller below into Super IOB 1.5. RUN Super IOB but do not format the target disk. Super IOB will copy tracks $3-$22 to the new disk.

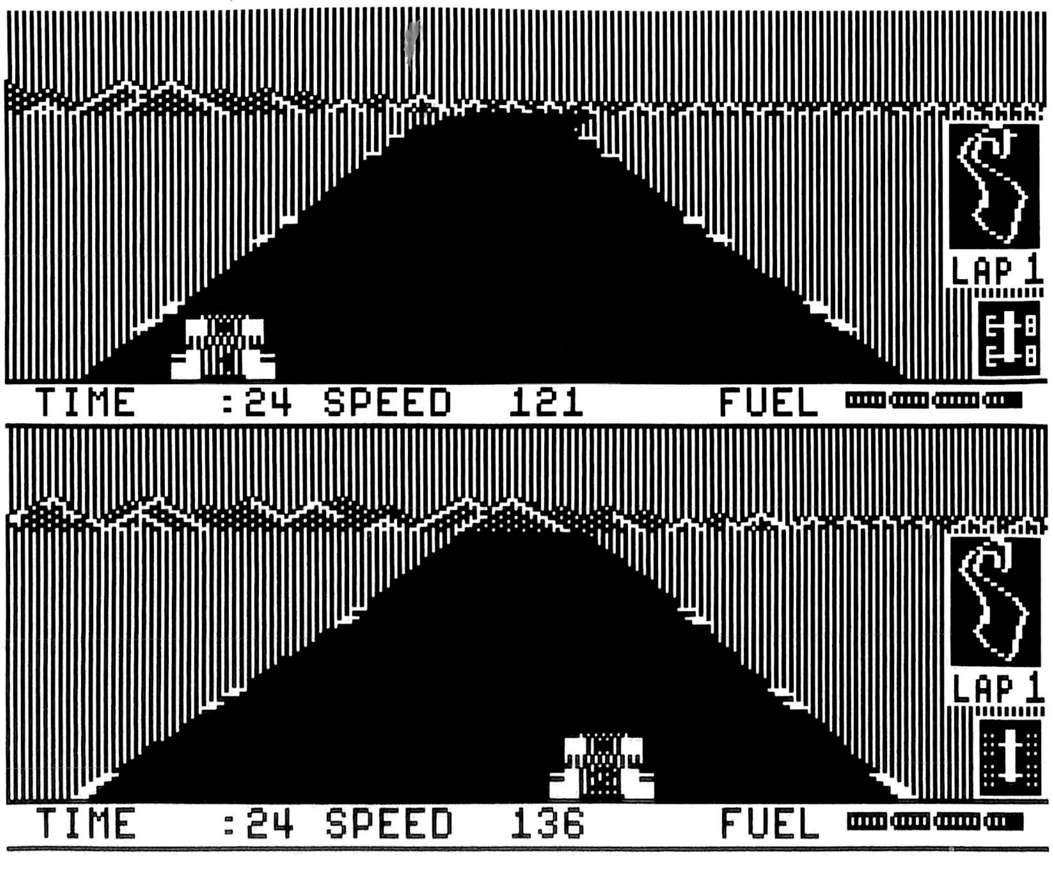
5) Have fun with your COPYAable Pitstop II.

controller

```
1ØØØ REM PITSTOP II CONTROLLER
1Ø1Ø TK = 3 : LT = 35 : ST = 15 : LS = 15 : CD = WR : FAST
     = 1
1Ø2Ø GOSUB 49Ø : RESTORE : GOSUB 17Ø : GOSUB 61Ø
1Ø3Ø GOSUB 49Ø : GOSUB 23Ø : GOSUB 61Ø : IF PEEK
     (TRK ) = LT THEN 1Ø5Ø
1Ø4Ø TK = PEEK (TRK ) : ST = PEEK (SCT ) : GOTO 1Ø2Ø
1Ø5Ø HOME : PRINT "THAT'S^ ALL^ FOLKS" : END
5ØØØ DATA ^ 255 ,255 ,255 ,255
```

controller checksums

```
1ØØØ  - $356B     1Ø4Ø  - $43C5
1Ø1Ø  - $2445     1Ø5Ø  - $8445
1Ø2Ø  - $F471     5ØØØ  - $3B18
1Ø3Ø  - $A55A
```



readers' softkey & copy exchange

Ken Black's softkey for...

Apventure to Atlantis

Synergistic Software
830 N. Riverside Dr.
Suite 201
Renton, WA 98055

Requirements:
At least 48K
A blank disk
COPYA or similar
A sector editor

It seems that this game is still rather popular. It offers good action, and requires some thinking to complete. This game requires some time to complete especially at the harder levels. Since using originals is a no-no we shall make a backup copy.

My attempts with a bit copier lead to bad copies time after time. Thank goodness it's easier to softkey than bit copy.

The Protection

It seems that the only protection on this disk is a protected DOS. The address and data epilogues have both been altered to E2 AA. The DOS appears to be an abbreviated form, in which only the RWTS and a few commands exist. This is probably because the disk is very full of program data. To put a full size DOS on the disk would require making room in extra tracks, and the catalog. The easiest softkey is to simply copy the disk with COPYA and fix the DOS so that it reads normal epilogues. All else on the disk is fairly normal. Let's press on and backup this popular game.

The Procedure

1) Boot a normal DOS.

2) Defeat DOS's error checking so that it will ignore the different epilogs.

```
CALL-151
B942:18
```

3) RUN COPYA and copy the disk.

4) Reboot DOS and start up your trusty old sector editor.

5) Perform the following sector edits to your new backup.

Track	Sector	Byte	From	To:
0	$3	$35	$E2	$DE
0	$3	$91	$E2	$DE
0	$2	$9E	$E2	$DE
0	$6	$AE	$E2	$DE

Your disk should now boot and work properly. Put the original away and use the copy with peace of mind.

Ken Black's softkey for...

```
THE
SERPENT'S STAR
Enter S to START, R to RESTORE a game
or D for a Demonstration.
```

Ultrasoft, Inc.
2509 152nd Ave, N.E., Ste. E
Redmond, WA 98052

Requirements:
Apple][
COPYA
A sector editor
two blank disk sides

Serpent's Star is one of the animated adventure games from Ultrasoft Inc. The reason I got it was that I had their other release "Mask of the Sun", and liked it very much. One of the things I noticed when I opened the package was that it had a note in it that said "DO NOT WRITE PROTECT". The protection scheme reads and writes to the disk as part of the protection. One note that cannot be overstated is that this program (and any program that alters the disk's surface) needs to be backed up to prevent damage to your disk drive head. You heard right, the disk drive head. Get your disk and hold it up to the light and turn the disk inside the jacket. At one point on the disk you will see a hole in the disk. The protection writes to the known location of the hole and if valid data is read back from that location, it knows that the hole isn't there. In other words, it expects a bad disk, or it won't boot. Part of my worries were concerned with that hole in the disk and what it would do to the read/write head on my disk drive. Enough passes of this rough surface on a read/write head will cause problems after a while. Since Howard Hughes I'm not, and service is expensive, I beg you to transfer this disk as soon as possible to a normal one. Disk drive parts aren't cheap!

Besides the hole in the disk, it seems that only the epilog bytes have been changed. Of course, the little message enclosed with the program telling about the protection scheme lets us know that there is more involved. We will copy the disk with COPYA, ignoring errors of course, and disable the test for the hole.

The Procedure

1) Boot normal DOS 3.3.

2) Enter the monitor and defeat DOS's read error checking.

```
CALL-151
B942:18
```

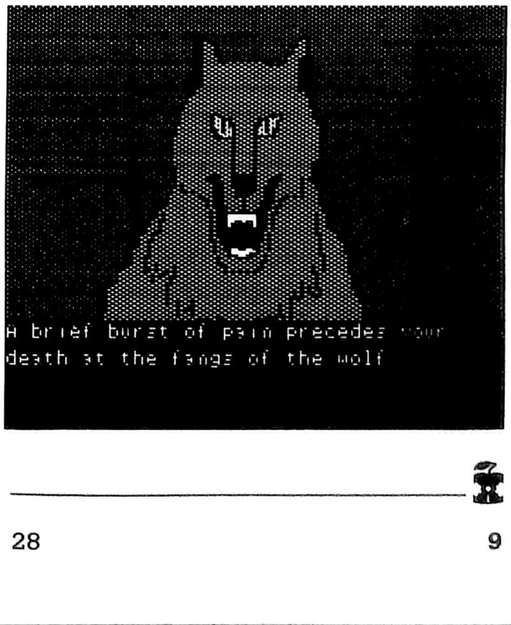
3) Get out your disk with COPYA and RUN it.

4) Copy both sides of the disk to your blank.

5) Get out your sector editor and make the following changes to the boot side.

TRACK	SECTOR	BYTE	FROM	TO
0	3	42	38	18
2	C	41	9D	EA
2	C	42	8F	EA
2	C	43	C0	EA

Now you should be able boot your copy without the write protect notch covered. It seems that somewhere in the file "LL(V27)" there is still something that checks for the read/write protection. This way works well for me, as the save game feature still works, while it may not in other backup procedures. You should notice that if it is write protected the disk will boot over and over. Good luck with the search for the missing scrolls.


```
A brief burst of pain precedes your
death at the fangs of the wolf
```

softkey for...

Ultima IV

by Mike Roetman

Ultima IV
Origin Systems, Inc.
34Ø Harvey Road
Manchester, NH Ø31Ø3

Requirements:
Apple][with 64K
Ultima IV
Super IOB v1.5

The crunch of sword against Orc bone is heard again in the realm of Lord British. The adventurers have returned to the largest, most sophisticated Ultima yet encountered. "ULTIMA IV: Quest of the Avatar" is a fun game to play, and a fun game to deprotect.

The Protection

Only Side A of Ultima IV has been protected. The other three disk sides can be backed-up with COPYA, and are accessable with normal DOS 3.3. After implementing this softkey the program disk will be just as open.

Origin Systems employs several schemes familar to most readers of COMPUTIST. The address prologues are changed and are different on almost every track. On most tracks the data prologues are a non-standard D5 AA B5. The address epilogues are altered and change sector by sector; the pre-address field sync bytes are DB instead of FF; and, the RWTS translate tables are not normal.

Normal RWTS Translate Tables

Not all possible byte values can be read by the disk drive. Hardware restrictions limit the number of valid eight-bit bytes on the disk to only 66, two of which are reserved. In order to get around this limitation, DOS breaks up the 256 bytes to be stored in a sector into 342 bytes in a process called nibbilization. Each of these 342 bytes has one of only 64 possible values from $ØØ to $3F. Since these 64 nibbilized values are not the 64 disk readable values, DOS uses the write translate table to convert the nibbilized data to disk valid bytes. For example, if the nibbilized byte is $ØØ the value $96 is actually written on the disk.

That which is coded must be decoded, and DOS has made provisions for this in the form of the read translate table. This table is used to convert disk bytes back to nibbilized form. To continue our example, the disk value of $96 is translated by DOS back to $ØØ.

Altered RWTS Translate Tables

An altered, protected DOS can be created by changing just one byte on each of the translate tables. For instance, changing the first byte of the Write Table to $AA will set $ØØ = $AA instead of $96. To read this disk properly the 21st byte (decimal) of the Read Table must be set equal to $ØØ, making $AA = $ØØ instead of $AA. The Read Translate Table value $96 = $ØØ does not have to be changed. The value $96 will never be read from the altered disk because it will never be written to the altered disk. Normal DOS will make no sense of this disk's data when it translates $AA as $AA.

There are two tipoffs that Ultima IV may use an altered translate table. First, when snooping through the original disk using The CIA program with altered sector marks, some of the sectors have invalid DATA checksums. Second, in track $11 a few sectors are normal, empty catalog sectors. Other sectors appear as outlines of normal catalog entries written in gibberish. Both of these findings can be caused by table alterations.

Next, it must be determined exactly what change has been made in the translate table. Sometimes the translate table is visible in the DOS, and it becomes an easy matter to find out what has been done by comparing it to the normal table. Unfortunately, Ultima IV's DOS is so heavily encrypted that this method does not work.

Random substitution can be tried. This is not as much of a longshot as it first appears. Since AA and D5 are valid (but reserved) bytes, most such changes involve substituting either AA or D5 for a normal value in the table. With two values to try in each position of the table, there are 128 likely changes to look for. A formidable task, but not impossible. The CIA's "Code Breaker" utility is a great help in such an undertaking. Success is determined by valid checksum reading of the disk and a normal catalog.

A more scientific method is to follow the steps that DOS uses to encode the data and find the variance. Sector $11/$ØE of the Ultima IV program disk looks like a "funny" catalog sector. If this were a normal catalog sector the first four bytes would be "ØØ 11 ØD ØØ". The nibbilized version of these normal bytes is basically 'Ø4 Ø7 Ø3 ØØ". Exactly how to nibbilize the data is beyond the scope of this article; "Beneath Apple DOS" by Worth and Lechner explains the procedure. The nibbilized data is normally converted to "9D A6 9B 96" by the write translate table and written on physical sector $11/$Ø4 starting at the 87th byte (decimal) after the data prologue. A raw nibble dump of the protected disk shows that "9D A6 D5 96" is written here instead. This means that nibbilized value $Ø3 is being translated as $D5 instead of $9B by the Ultima IV write translate table and $D5 must become $Ø3 on the altered read translate table.

The IOB

Sometimes the old methods work the best. The original SUPER IOB simply ignored epilogue bytes in subroutine 17Ø. SUPER IOB v1.5 requires that the altered epilogue values be entered as DATA statements. Since the address epilogue in Ultima IV changes with each sector (512 changes on the disk), it is easier to just ignore the epilogue bytes. A long list of DATA statements is still needed because the address prologue changes track by track.

To compensate for the altered Translate Tables the D5 = D5 on the Read Translate Table must be changed to D5 = Ø3. This is accomplished by the POKE 47829,3 in the controller. POKE 47829,213 normalizes the table. The write translate table does not need to be changed because we are never writing to the original disk.

Clean Up Details

After you complete your copy, patch a DOS onto the disk. A fast DOS is preferable. Write an Applesoft HELLO program that ends with the statement:

```
PRINT CHR$(4 ) "BRUN I^A NIT"
```

and save this to the new disk.

A few modifications are still needed to make the normal RWTS routine fully compatible with Ultima IV. To accomplish these BLOAD S^A UBS, make the changes found in table 1, and BSAVE S^A UBS, A$0800, L$1800. Or change these values directly on the disk at Track $12, Sector $09 with a sector editor.

Have Fun

Ultima IV is a pleasure to play. It should be in every adventure gamers library. Buy it, back it up, enjoy it.

And when you are playing your next "Copy the Disk" game, don't forget about the translate tables. They may hold the key you are looking for.

Step by Step

1) INITialize a disk preferably with a fast DOS.

INIT HELLO

2) Install the controller at the end of this article into your copy of Super IOB v1.5 and RUN it.

3) Type in this short program.

```
10 PRINT CHR$( 4) "BRUN I^A NIT"
```

4) Save this program as HELLO to your copy of Ultima IV.

SAVE HELLO

5) Load in the file called S^A UBS.

CALL -151
BLOAD S^A UBS

6) Make the changes found in the following table to the file.

Table 1

LOCATION	FROM	TO
0A9F	8D	EA
0AA0	5D	EA
0AA1	BD	EA
0AA4	8D	EA
0AA5	2C	EA
0AA6	BF	EA
0AA8	E8	B7
0AAA	B7	E8

7) Save this modified program to your disk.

BSAVE S^A UBS, A$0800, L$1800

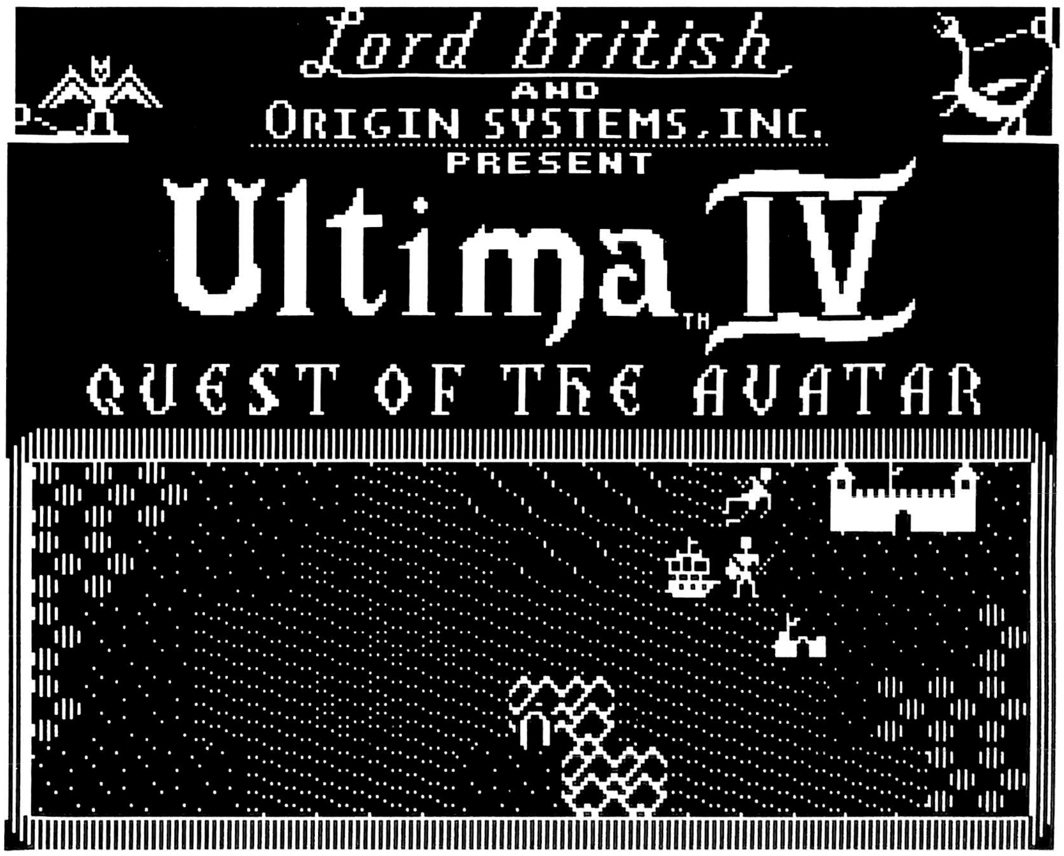
That's it!

controller

```
1000 REM ULTIMA IV
1010 TK = 3 :LT = 4 :ST = 15 :LS = 15 :CD = WR
1020 POKE 47405 ,24 : POKE 47406 ,96 : POKE 47497
     ,24 : POKE 47498 ,96
1030 POKE 47829 ,3 :T1 = TK : GOSUB 490 : GOSUB 210
1040 GOSUB 190 : GOSUB 610
1050 TK = TK + 1 :LT = LT + 1 : IF PEEK (BUF ) < MB
     AND TK < 35 THEN 1040
1060 POKE 47405 ,208 : POKE 47406 ,19 : POKE
     47497 ,208 : POKE 47498 ,183 : POKE 47829
     ,13 : GOSUB 230
1070 TK = T1 :LT = 35 : GOSUB 490 : GOSUB 610 : IF
     PEEK (TRK ) = LT THEN 1090
1080 TK = PEEK (TRK ) :ST = PEEK (SCT ) :LT = TK
     + 1 : GOTO 1020
1090 HOME : PRINT "COPYDONE,^ DOS^ NOT^
     COPIED." : END
5000 DATA 213 ,170 ,181
5010 DATA 215 ,170 ,151
5020 DATA 213 ,170 ,150
5030 DATA 213 ,170 ,151
5040 DATA 215 ,170 ,150
5050 DATA 215 ,170 ,151
5060 DATA 221 ,170 ,158
5070 DATA 221 ,170 ,159
5080 DATA 213 ,170 ,181
5090 DATA 223 ,170 ,158
5100 DATA 223 ,170 ,159
5110 DATA 221 ,170 ,158
5120 DATA 221 ,170 ,159
5130 DATA 223 ,170 ,158
5140 DATA 223 ,170 ,159
5150 DATA 213 ,170 ,150
5160 DATA 213 ,170 ,181
5170 DATA 213 ,170 ,151
5180 DATA 215 ,170 ,150
5190 DATA 215 ,170 ,151
5200 DATA 213 ,170 ,150
5210 DATA 213 ,170 ,151
5220 DATA 215 ,170 ,150
5230 DATA 215 ,170 ,151
5240 DATA 213 ,170 ,181
5250 DATA 221 ,170 ,158
5260 DATA 221 ,170 ,159
5270 DATA 223 ,170 ,158
5280 DATA 223 ,170 ,159
5290 DATA 221 ,170 ,158
5300 DATA 221 ,170 ,159
5310 DATA 223 ,170 ,158
5320 DATA 213 ,170 ,181
5330 DATA 223 ,170 ,159
5340 DATA 245 ,170 ,182
5350 DATA 245 ,170 ,183
5360 DATA 247 ,170 ,182
```

controller checksums

1000	- $356B	5140	- $690B
1010	- $3189	5150	- $0DF7
1020	- $C562	5160	- $9708
1030	- $545E	5170	- $E228
1040	- $DDB4	5180	- $B79F
1050	- $A5C8	5190	- $B024
1060	- $46F2	5200	- $AD98
1070	- $DE82	5210	- $CCE4
1080	- $3CF1	5220	- $2D67
1090	- $8C6A	5230	- $CE20
5000	- $7F44	5240	- $D255
5010	- $A7D0	5250	- $591D
5020	- $3759	5260	- $446D
5030	- $A9EE	5270	- $97AC
5040	- $355C	5280	- $DCF9
5050	- $C914	5290	- $1B35
5060	- $E02F	5300	- $4A59
5070	- $BBA0	5310	- $E57C
5080	- $F498	5320	- $199F
5090	- $F49E	5330	- $0D68
5100	- $FE8E	5340	- $BEF8
5110	- $FA81	5350	- $3174
5120	- $994B	5360	- $8E91
5130	- $4834		



ROBOT ODYSSEY I

By Steve Luzietti

The Learning Company
545 Middlefield Road, Suite #170
Menlo Park, CA 94025
$49.95

Requirements:
64K Apple][Plus and up
COPYA
A sector editor
A double-sided blank disk
(or two single sided blank disks)
Robot Odyssey I

Robot Odyssey is an excellent educational program from the same company that brought us Rocky's Boots. It offers instruction encompassing electrical engineering fundamentals, digital logic, and circuit design all in a neat, easy-to-use and entertaining hi-res package! Unfortunately, (or fortunately for us challenge-seeking deprotectors) the program does have one major drawback: copy protection, and a sneaky one at that.

The disk itself is formatted in a slightly altered DOS 3.3 format. The only modifications are that both the address field and data field epilogues are changed from DE AA to FF FF. This can be easily defeated by turning off DOS 3.3's epilogue checksum, B942:18 in the monitor (I'm sure by now you've seen this so many times that you probably recite those mystical numbers in your sleep); reading from the disk in this altered format; turning the epilogue checksum back on, B942:38 in the monitor; and writing to your copy disk in standard format. This normalization process can be easily accomplished with a slightly modified COPYA.

If you boot the normalized copy of Robot Odyssey, everything seems to function properly. The disk boots fine, the program loads its different portions correctly, and everything seems OK until you try to use the soldering iron. The soldering iron will not work! This is really a nasty trick done by the protectors, since the operation of the soldering iron is an integral part of the program. This "malfunction" clues us in on another bit of protection, probably a check done by the program for the original disk, either during the boot or during the loading of a program segment. After booting the disk and breaking into the monitor during the title page display, I found some suspicious-looking portions of code in memory pages $5D and $62: two disk-checking routines.

The disk-checking routines at memory locations $5D4F-5DD5 and $6268-62EE are essentially the same. The code looks for certain bytes on the disk and if these bytes are not found, the code branches to an "error handler" (for lack of a better term) at $5DCE for the first routine or at $62E7 for the second routine. By analyzing the exiting conditions of the disk check, one may notice that if the disk checks out correctly, the carry bit is set, but if the disk check fails, the carry bit is cleared. For illustration purposes, here is a disassembly of the error handler code for the first routine.

This portion of code will be executed if the disk checks out correctly. Note the SEC (set carry) and the branch to $5DD3:

```
5DCB-   38          SEC
5DCC-   BØ Ø5       BCS   $5DD3
```

However, in the actual disk-accessing portion of the disk check, if certain bytes are not found on the disk, the program will branch to the following code segment. In other words, the following code will be executed if the disk check fails. Note the CLC (clear carry):

```
5DCE-   C6 Ø9       DEC   $Ø9
5DDØ-   DØ 97       BNE   $5D69
5DD2-   18          CLC
5DD3-   BD 88 CØ    LDA   $CØ88,X
5DD6-   6E FE Ø3    ROR   $Ø3FE
5DD9-   4C ØØ 4Ø    JMP   $4ØØØ
```

The only important condition on exiting from the disk check is the carry bit. This is the key for successfully cracking Robot Odyssey. We must change the instructions at $5DD2 and $62EB (the latter is in the error handler for the second disk checking routine) from CLCs to SECs. So, in cookbook fashion:

1) Load in COPYA.

LOAD COPYA

2) Add the following lines so that COPYA will ignore the address and data epilogues when reading but use them when writing.

199 POKE 47426,24
249 POKE 47426,56
259 POKE 47426,56

3) Type RUN and let COPYA load its object file. Copy both sides of the Robot Odyssey disk with the modified COPYA.

4) Get your sector editor up and running and perform the following sector edits to both sides of the copy disk:

```
Track  Sector  Byte  From  To
-----  ------  ----  ----  ---
 $ØØ    $ØF    $D2   $18   $38
 $Ø4    $ØE    $EB   $18   $38
```

That's it! You now have a completely cracked and operational copy of Robot Odyssey.

softkey for...

Rendezvous

By The Nipper

Rendezvous
Peachtree Software
3445 Peachtree Road N.E.
Atlanta, GA 30326

Requirements:
Apple][with at least 48K
FID from the System Master

Rendezvous is one of the high quality educational programs previously handled by Eduware that is now distributed by Peachtree. It is a simulation of the space shuttle takeoff and docking procedures and is an excellent way to introduce students to the laws of physics involved in space flight.

Unfortunately Peachtree has installed a new protection system that makes backing up Rendezvous very difficult. Further, Peachtree does not supply a backup, although they do offer a replacement policy. As any of you who work with children know, the first rule of computers is never give your original to a child. The difficulty in backing up Rendezvous led me to look for a method of deprotecting it.

The Deprotection

The first thing that I noticed was the small slip of paper that came with the program that said it would not boot if a language card was present. Sure enough, it wouldn't. So I took out my language card and booted up Copy][+ 5.0 to have a look at the address and data prologues and epilogues. The same thing can be done with any good nibble editor. The epilogs had been changed from DE AA to FF FF. I then booted up my sector editor and set it to ignore the epilogs and checksums. Everything went well until I tried to read any track other than zero. All I got was the dreaded I/O Error. I then scanned the disk using the hi-res scan feature (like in Locksmith 5.0). You could see extra white dots between the standard sync fields. Checking with the nibble editor showed sync bytes within the data fields. This is a sure indicator of a modified translation table. Since I couldn't capture it with my Integer Language card, I tried to boot the disk into the extended 80-column card (see "Secret Weapon: Ramcard" by Ken Greenlaw, COMPUTIST No. 16), but the program checked for that and wouldn't boot.

Frustrated, I went off to bed dreading the thought of having to boot code trace the program to remove the protection. In the morning I remembered that any disk that uses modified RWTS does so only after sectors 0 to 9 have been read in since ROM sets up a temporary set of tables, which is replaced by information on these sectors. With my sector editor patched, I read in track 0, sector 4 and looked at the nibble translate table which occupies bytes $29 to $68. This is the table which is used to translate data in RAM to the disk.

Instead of finding the standard start of 96 97 9A I found AA 97 9A. A quick look revealed no other changes. So, the data had been written to the disk substituting AA for 96. The byte translate table occupies bytes $96 to $FF and is used to translate the bytes on the disk back into their original form in RAM. The value at $AA had been changed to $00 to match the change in the nibble translate table. The method for removing this protection was obvious. All that needed to be done was to modify a standard byte translate table (reading) to match the one used for Rendezvous while leaving the nibble translate table (writing) standard. Then if we patched DOS so that it ignored the epilogues, we could copy Rendezvous onto a standard disk.

The Softkey

1) Boot your DOS 3.3 system master.

2) Remove your system master and put in a blank disk. Initialize the disk with the name of the boot program on Rendezvous.

INIT EDU-WARE

3) Delete the boot program.

DELETE EDU-WARE

4) Put your system master back in the drive and run FID.

BRUN FID

5) Stop FID by pressing Reset.

6) Enter the monitor and alter the byte translate table so that it matches the one in Rendezvous.

```
CALL-151
BAAA:00
```

7) Disable the address epilog check.

B988:18 60

8) Disable the data epilog check.

B925:18 60

9) Restart FID by calling its starting address.

803G

10) Copy all the files from the Rendezvous disk onto your initialized disk with FID. You should now have a broken version of Rendezvous.

You will find that a number of the other Edu-Ware disks that are handled by Peachtree are handled in exactly the same way. I have found this to be the case with Introduction to Counting, Spelling and Reading Primer, and Spelling Bee Games. Note that the files on Spelling Bee Games are too large to fit on a standard disk, so you will have to patch your initialized disk to free up the sectors on track 2 that DOS does not use. Change track $11 sector 0, bytes $40 and $41 from $00 $00 to $FF $E0.

Enjoy.

Word Attack

by Dave Stanton

Davidson & Associates
Groveoak Place #12
Rancho Palos Verdes, California 9Ø274
$49.95 each

Requirements:
Apple][Plus with a way to reset into the monitor; or 128K //e, or //c.
Blank disk(s)
Super IOB v1.5
Classmate, or Word Attack!

Davidson & Associates offers well designed and effective software for the school and home market. Instead of the dazzling but often superfluous high-resolution graphics offered by many educational publishers, this company combines sound teaching principles with a truly user-friendly programing style. The result is a simple elegance that makes it easy for students and teachers to become comfortable with the software.

For this achievement Davidson's offerings have received significant and just recognition, but in one way the company has failed to keep up with the pack. While many publishers now include a backup or a one-backup copy-program with their disks, Davidson still follows the old practice of charging extra for a backup, which must be ordered by mail. Teachers, perhaps better than others, know the problems this system can cause. If a teacher is to entrust his students' grades to "Classmate," for example, he must be certain that his data will not become inaccessible because of inadvertent damage to his original-- not even for a short time.

In the interest of providing peace of mind to the purchasers of Davidson software, we embark on the adventure of deprotecting "Word Attack!" and "Classmate." No doubt the few hints offered here will provide encouragement and support to those who need to deprotect other programs from the same publisher.

The Reconnaissance:

The logical first step in backing up any program is to try COPYA. If that fails, commercial copy programs may work. Neither of these approaches creates a functional copy of "Classmate" or "Word Attack!"

Next, one might try to halt the execution of a program with CTRL C or Reset so that the code, whether in Applesoft or machine language, can be analyzed and/or saved to a standard DOS disk. Sometimes opening the disk drive door at the right time works. Most copy-protected software uses a modified DOS which prevents effective breaking of the program with these methods. Both "Classmate" and "Word Attack!" follow this tradition.

How to load the protected DOS and stop its execution at will- that is the question. A copy card like Wildcard or Snapshot provides one answer, but we will use the method suggested in Ken Greenlaw's article "Secret Weapon: RAMcard" (Hardcore COMPUTIST No. 16). To use this procedure, you must have an Apple //c or a //e with an extended 8Ø column card. You will also need the programs XFER.BOOT and RESTORE that this method uses. If you have 128K, type in the XFER.BOOT hexdump (included here) and save it:

BSAVE XFER.BOOT,A$3ØØ,L$1A

Then type in the RESTORE hexdump and save:

BSAVE RESTORE,A$3ØØ,L$21

Using Greenlaw's procedure does, in fact, allow us to capture and save the altered DOS from "Classmate," and the same one will work with "Word Attack!" and probably the other programs from Davidson.

After you have INITed a disk with the proper boot program, Super IOB 1.5 with the swap controller will put the deprotected files on your disk. You can now catalog the disk and load each of the files, most of which are written in Applesoft BASIC.

Unfortunately, you will find that the programs do not run. Hidden in the code are a few lines that check for the original operating system. Those checks are simple but effective. We need to eliminate them, a simple process under normal conditions.

However, you will find that many of the files on the "Classmate" and "Word Attack!" disks will not list after you have loaded them. Instead you will get Ø REM, and nothing more. The programer has incorporated some type of control character into a few of the early REMs. That character terminates the list when encountered. By eliminating those lines entirely, you will be able to list and analyze the remaining lines.

Now to find the protection lines. Unusual PEEKs and CALLs are suspect, especially since most of the files are written in BASIC. It is good deprotection strategy to check your "Apple II Reference Manual" to determine what each of the suspect lines does. The culprits must be expunged. As you will see in the procedure that follows, some of the files require no line deletions, while others require two or three such changes.

Once the necessary changes have been made to each file, simply unlock the file on your copy and save the revised version over it.

You may notice that WORD ATTACK! HELLO, a file on "Word Attack!," has a very unusual list:

```
1Ø POKE 1Ø4,32: RUN
65355 REM
65355 REM
```

No changes need be made to this file but if you wish to list it, POKE 1Ø4,32 in immediate mode. This sets the program pointer to the proper starting address. Now you can LIST normally. There is, however, another program hidden inside this one. Finding it offers a

or Classmate

challenge, and although you need not understand this potential enemy weapon to copy "Word Attack!," you may find the same technique used on other disks.

The Attack

NOTE: In each step that requires deleting lines Ø and 1Ø, you must delete the line before listing or running.

1) INIT a new disk with "WORD ATTACK! HELLO."

INIT WORD ATTACK! HELLO

For Classmate:

INIT CLASS

2a) If you have a copy card or other way to pop into the monitor, then follow this step. Else, go to step 2b. Boot your original, and when the drive stops, stop the program. While in the monitor, Move the RWTS to safe memory:

19ØØ<B8ØØ.BFFFM

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

We now return you to our regularly scheduled article.

2b) Assuming you have 128K, Boot a normal DOS. Load in XFER.BOOT.

PR#6
BLOAD XFER.BOOT

3) Put your original disk into the drive and boot it into upper memory.

PR#3
CALL768

4) When the drive stops, hit Control-Reset. Go into the monitor to prepare for RESTORE and load RESTORE.

CALL-151
3F8: 4C ØØ Ø3
BLOAD RESTORE

5) Move the trapped RWTS into lower memory at $19ØØ.

19ØØ<B8ØØ.BFFF⌃Y

6) Save the protected RWTS to the disk on which your Super IOB 1.5 is stored.

BSAVE
RWTS.DAVIDSON,A$19ØØ,L$8ØØ

7) Load Super IOB 1.5 and enter the swap controller listed below. RUN to make your copy.

8) LOAD WORD ATTACK! and delete line Ø. Enter line 7 as shown. UNLOCK WORD ATTACK! and SAVE WORD ATTACK!

```
7 PRINT CHR$( 4 ); ""MAXFILES 1'': GOSUB 217Ø:
    GOSUB 778Ø: GOSUB 738Ø
```

9) LOAD WORD ATTACK! EDITOR. Delete line #5. UNLOCK WORD ATTACK! EDITOR, and SAVE WORD ATTACK! EDITOR.

10) Copy the "Word Attack!" data disk with COPYA.

The Attack of "Classmate"

1) Follow the steps for "Word Attack!" from step 1 to step 7.

2) Load CLASS and delete lines Ø, 1Ø, and 4Ø.

3) Change line 25Ø as shown, and save the modified file.

25Ø PRINT CHR$(4);'RUN CLASSMATE1'
UNLOCK CLASS
SAVE CLASS

4) Load CLASSMATE1 and delete lines Ø, 1Ø, and 35. Save the new version of CLASSMATE1 to the disk.

5) Load CLASSMATE2 and delete lines Ø, 1Ø, and 3Ø. Save the modified CLASSMATE2 to disk.

Victory:

You should now have a perfectly functioning, deprotected copy of "Classmate" and/or "Word Attack!" As a bonus, you have a good start at deprotecting other programs from this publisher.

XFER.BOOT and RESTORE are reprinted from Issue No. 16 for your convenience. Refer to the original article for an explanation of what they do.

XFER.BOOT

```
Ø3ØØ: 2Ø 58 FC 8D 55 CØ 8D 5F    $FEØ2
Ø3Ø8: CØ A9 ØØ 8D ED Ø3 A9 C6    $FDDA
Ø31Ø: 8D EE Ø3 18 69 8Ø 38 4C    $ACDC
Ø318: 14 C3                      $C571
```

RESTORE

```
Ø3ØØ: 18 2Ø 11 C3 8D Ø9 CØ A6    $44AB
Ø3Ø8: 67 A4 68 8D Ø8 CØ 86 67    $9653
Ø31Ø: 84 68 8D Ø9 CØ A6 AF A4    $D83Ø
Ø318: BØ 8D Ø8 CØ 86 AF 84 BØ    $6D11
Ø32Ø: 6Ø                         $ABDD
```

controller

```
1ØØØ REM DAVIDSON CONTROLLER
1Ø1Ø TK = 3 : LT = 35 : ST = 15 : LS = 15 : CD = WR : FAST
    = 1
1Ø2Ø GOSUB 36Ø : GOSUB 49Ø : GOSUB 61Ø
1Ø3Ø GOSUB 36Ø : 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
1ØØ1Ø  PRINT  CHR$  (4  )  "BLOAD^
    RWTS.DAVIDSON,A$19ØØ"
```

controller checksums

```
1ØØØ  - $356B      1Ø4Ø  - $ØD4F
1Ø1Ø  - $2445      1Ø5Ø  - $E4ØC
1Ø2Ø  - $C9Ø8      1ØØ1Ø - $DCB4
1Ø3Ø  - $FB9C
```

Fingerprint

by Michael Stafford

Thirdware Computer Products
4747 NW 72nd Avenue
Miami, FL 33166

One of the biggest problems for readers of COMPUTIST magazine who own newer machines like the //e, is in finding an easy way to reset into the monitor from protected software. Many solutions have been offered, ranging from the exotic - burning your own custom ROM chips, to the more mundane - non-maskable interrupt (NMI) cards (like Wildcard, Replay, etc.), or forcing a boot into the auxiliary RAM on 128K machines. I'd like to offer you yet another solution, which is essentially free - if you need a printer interface that is. It's called the Fingerprint Plus printer interface, manufactured by Thirdware Computer Products.

I've been using this product for over six months now, and have been very impressed by it's performance. First of all, it has both parallel and serial graphics capable interfaces on the same card, enabling you to run both a dot matrix and a letter quality printer off the same interface. In fact, you can print with both simultaneously if you like. The parallel portion of the board is Grappler+ compatible, and the serial portion is Apple Super Serial compatible. This insures almost universal software compatibility, no matter which output you choose. I've been using the interface with a Panasonic 1091 parallel printer, and haven't encountered any programs it did not function perfectly with, including Dazzle Draw, Fontrix, and even Mousepaint - although Mousepaint does require activating a special ROM routine first.

All these features are very nice, but the thing that really distinguishes this interface from the rest of the pack is an item that Thirdware calls an "activator" button. This button is a 1x1 inch flat wafer switch that is attached to the interface by a paper thin, mylar ribbon cable. You stick this adhesive backed switch on the case of your Apple, and thread the cable between the computer case and cover. The cable is so thin it doesn't interfere with the closing of the cover at all.

Just as with any non-maskable interrupt card, pushing this button causes the program in memory to "freeze" - but there the similarity ends. Having pushed the button, the menu to the right appears on your screen.

As you can see, the menu is divided into two main parts. The smaller section, across the top, gives the main functions of the Fingerprint Plus: Display, Print, Typewriter, and Jump. The larger section at the bottom lists 30 yes-or-no options. You move your cursor back and forth between the Function and Option sections by pressing the space bar. The left and right arrow keys move your cursor around the screen within the Function and Option sections, and the return key toggles yes or no for each option.

At this point you have two main choices available - if all you want to do is print your current screen, you just press the return key, and whatever was on your screen at the moment you pushed the "activator" button will be printed. No matter if it's text, graphics, or mixed mode, it will automatically be dumped to your printer. If however, you want to manipulate the contents of your screen, or perform some other operation (like jumping into the monitor!) you do so thru the menu.

When you choose the Display function, you can preview/choose a particular screen available for printing: Hi-res pages 1 and 2, Text pages 1 and 2, mixed mode, etc.

After you've selected the screen you want, hitting the space bar returns you to the Print function. Press return at this point and whatever screen you have selected will dump to your printer.

The Print function does just that - lets you print what you have selected through either the options menu, or the Display function. Hitting the space bar from the Print function sends you into the option menu, where you can change any of the print defaults, select the type of output - parallel, serial, or both, or even change the interface card dip switch settings! Once you've made your option selections, hitting the space bar once again takes you back to the Print function, where pressing return will print using the options you've picked.

Selecting the Typewriter function allows you to either use the keyboard like a typewriter (a la Applewriter), sending output directly to your printer, or you can send control commands directly to the interface card or your printer. The Fingerprint Plus command set (included in the manual), is more extensive than the option menu, even allowing you to change the serial output baud rate from the keyboard.

Last, but certainly not least, is the Jump function. Once you've selected this function, you press the space bar to enter the option menu, and pick which of the two jump option

Plus

you want. The jump to FP RAM option activates any programs or routines that you had previously stored in the user-available RAM on the Fingerprint card. This can be a very powerful tool, limited only by the size of the RAM available, and your imagination. At the time of this writing, the software program to load your routines into this user RAM was still not finished, but Thirdware assures me it will be shipped shortly.

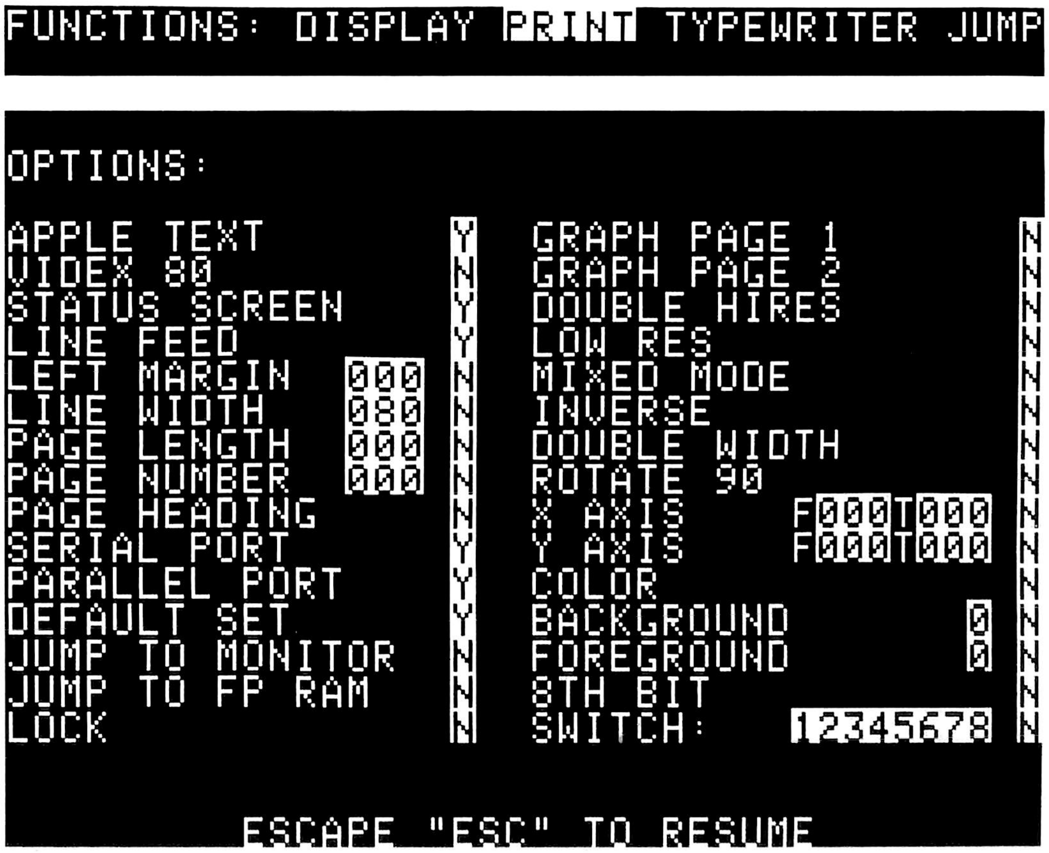
The jump to monitor option, quite obviously, does just that - and is one of the main reasons I purchased this interface over the myriad of others on the market. All you do is push the "activator" button to freeze the program, hit the space bar to get into the Option menu, select the jump to monitor option, hit the space bar again to toggle back to the Function menu, select the Jump function, hit return, and you're in the monitor. It sounds like a lot of steps, but it's actually quite quick and painless, and it certainly beats buying an NMI card.

If you have an external modem, or are thinking of acquiring one, it can be run off the serial output of the board. Thirdware sells a special cable for this purpose, the cost of which is, I believe, $28.00. If you are not using the serial port for your printer, this can save you having to buy another interface board, and free up another slot in your computer as well.

Thirdware deserves a few kudos for their technical support of this product. I have had occasion to call them a few times with some questions, and have found their representatives to be extremely knowledgeable, and willing to go to great lengths to help you out.

If there is anything I can find to criticize, it's the manual. All the information is there, but in spots it's rather sketchy, and on the whole it's crudely done. In fairness, I do have one of the earlier versions of the board, and it's possible the manual has since been improved.

Current list price for the Fingerprint Plus is $149.00, although it's commonly available mail order for $99.00, and for that price, it's one of the best hardware bargains around.



```
FUNCTIONS: DISPLAY PRINT TYPEWRITER JUMP

OPTIONS:

APPLE TEXT             Y   GRAPH PAGE 1            N
VIDEX 80               N   GRAPH PAGE 2            N
STATUS SCREEN          Y   DOUBLE HIRES            N
LINE FEED              Y   LOW RES                 N
LEFT MARGIN        000 N   MIXED MODE              N
LINE WIDTH         080 N   INVERSE                 N
PAGE LENGTH        000 N   DOUBLE WIDTH            N
PAGE NUMBER        000 N   ROTATE 90               N
PAGE HEADING           N   X AXIS        F000T000  N
SERIAL PORT            Y   Y AXIS        F000T000  N
PARALLEL PORT          Y   COLOR                   N
DEFAULT SET            Y   BACKGROUND           0  N
JUMP TO MONITOR        N   FOREGROUND           0  N
JUMP TO FP RAM         N   8TH BIT                 N
LOCK                   N   SWITCH:       12345678  N

          ESCAPE "ESC" TO RESUME
```

Beyond Castle Wolfenstein

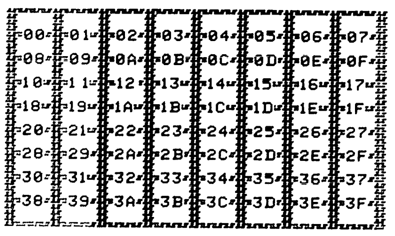
part2

by Lars Grant-West

Requirements:
A broken copy of Beyond Castle Wolfenstein (or a way into the monitor)
A desire to cheat

After reading through Ray Darrah's article on the deprotection of Beyond Castle Wolfenstein (see "BENEATH Beyond Castle Wolfenstein," COMPUTIST No. 13, pg. 12) and following his instructions, I managed to make an unprotected copy of the game. I tried all of his APTs, and became so fascinated by the possibilities that I decided to go on from there and do a bit of experimenting on my own. After hours of poking around, the only thing I learned was to stay as far away as possible from memory location $4309. Not being one to take notes, I also don't have even an inkling of how I managed to transform all the guards into roaming Hitlers, and I doubt I could do it again if I tried. After repeated failure at making any earth shattering revelations, I decided to leave such experimenting to the pros.

Diagram 2



00	01	02	03	04	05	06	07
08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17
18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27
28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37
38	39	3A	3B	3C	3D	3E	3F

Even though my searching didn't amount to much, I did come out of the experience with but a few battle scars and a set of maps and charts that might make a helpful supplement to Mr. Darrah's article.

Diagram 1: A map of the BEYOND CASTLE WOLFENSTEIN bunker. Although the appearance of the rooms may change, the basic layout of the bunker does not. As you can see, each square or room on the map contains two sets of values. The top number is that of the room itself. This has no bearing whatsoever on the function of the program, but can be used by you for reference. The bottom value is the hexadecimal number of the room, that which the computer understands. To move to another room, press Reset. You will find yourself in the monitor. As stated in Mr. Darrah's article, $4340 contains the room number. For example, to place yourself in the main guard room (#59), you would type (from the monitor):

```
4340: 3B
1284G
```

Or, if you're even more daring, you can visit the royal goose-stepper himself by replacing the 3B above with a 3C. Understand?

Diagram 2: Player room coordinates. This chart gives you all sixty-four locations that your spy can possibly go to in a room. Its use is similar to that of the bunker map from Diagram #1, but instead of going from room to room, you merely hop about the screen. To do this, you press Reset, and are, once again, in the monitor. To jump to, say, the upper left hand corner of the screen, you would type:

```
4343: 00
1290G
```

Experiment for a while. Room one is a good place for practice, but be careful. If you teleport to any location outside the room's wall boundaries and then walk off the screen you could end up...Oh, why should I tell you. You'll probably want to see for yourself, won't you. Just don't blame me if you never get back. (By the way, you can get into the same rather unsettling situation by trying to put in a room value that's not on the map.)

Well, back from the Twilight Zone. This teleporting is a pretty neat trick, and if it's not good for anything else, it certainly helps you keep out of sight of those big-mouthed desk clerks.

Diagram 3: This final chart gives the memory locations of all seventy-two wall blocks on the screen. As described in Ray Darrah's article, the blocks actually overlap, but in the name of what little sanity I have left intact, I decided to depict the blocks as being adjacent.

To the right of the chart, there is a list of all the useable values worth knowing. A "00" in any location blanks it out. A "01", on the other

Room 1

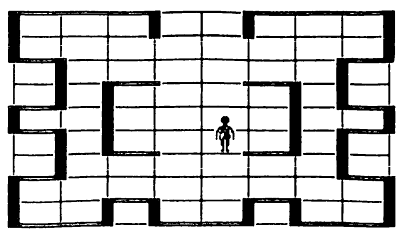
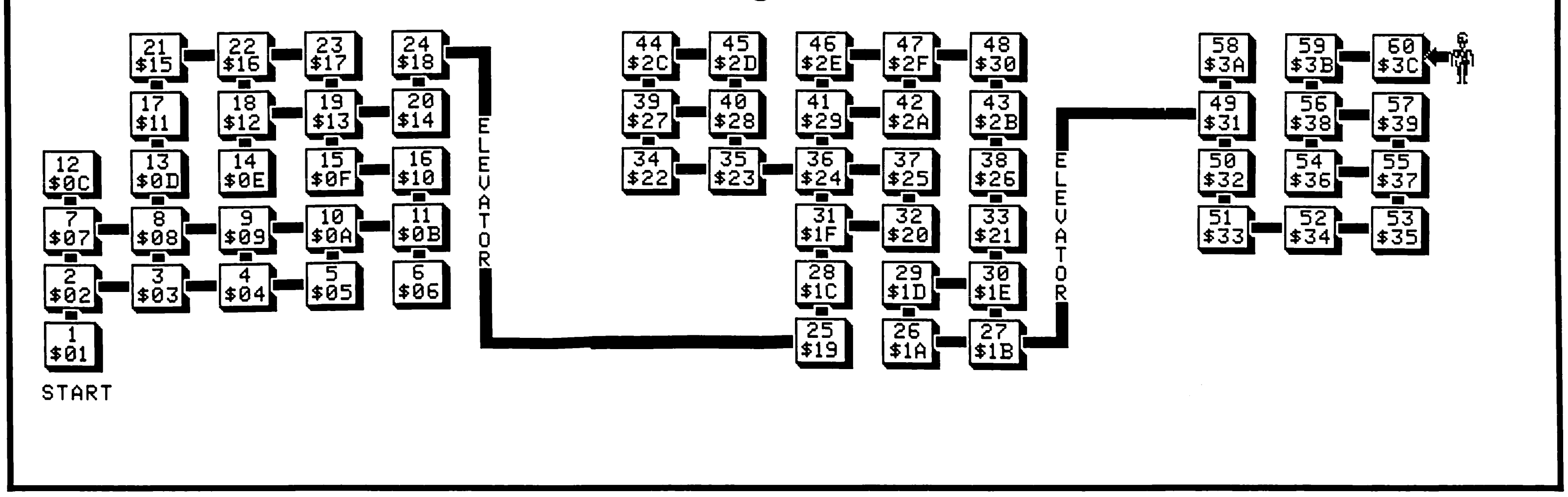
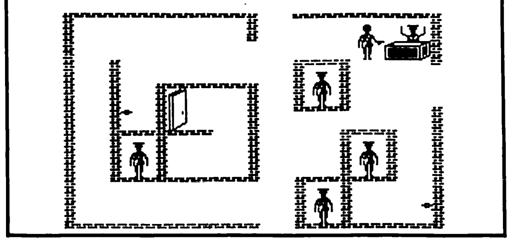


Diagram 1



hand, puts a wall on the left hand side of the block. There are a number of other values possible besides those listed, But using them results in the materializing of things such as "solid air", invisible, but tangible sections of nothing. Somewhere along the line I also remember discovering elevators that never failed to leave you in bizarre situations such as those described for diagram #2. (I also noticed that even on the bottom level of the bunker, these elevators always went down. The phrase "Abandon all hope" immediately flashed across my mind.)

Room 2



One of the more useful purposes that I put my new chart to was that of imprisioning guards, or at the very least, blocking off those dratted alarm buttons so guards cannot reach them. Say you have just entered a room with one guard walking back and forth in front of the only other exit, located at the top of the screen. You wait in the shadows until he has walked, unsuspecting, into the upper left corner of the screen (Diagram 4). You hit the RESET button before the he knows what hit him, and are thrown into the temporary safety of monitor. Looking at your screen block chart, you see that the guard is occupying locations $4000 and $4008. You put a $07 at $4000 for the top of the box to trap him in and a $0B for the bottom of the box at $4008.

4000: 07
4008: 0B
1290G

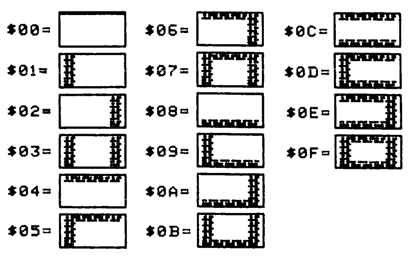
and the poor chap suddenly finds himself quite trapped, while you are free to walk by unheeded. (Note: For all you sadists out there, The walls of the guard's new "prison" are one-way (see below). Although you can't shoot the guard, you can pull out your dagger and run him through right through the wall. Just don't walk into the block, or you too will be trapped.)

As stated in the previous paragraph, due to the fact that the blocks overlap, all walls, except those at the edge of the screen, occupy portions of two adjacent blocks. A wall must be registered in both blocks for it to be completely solid. If you only register a wall in one block, it will only be solid from the side of the block you registered it on. Example: If, in an empty room, you wanted a wall between blocks $4013 and $4014, and only typed:

4013: 02
1290G

Diagram 3

4000	4001	4002	4003	4004	4005	4006	4007
4008	4009	400A	400B	400C	400D	400E	400F
4010	4011	4012	4013	4014	4015	4016	4017
4018	4019	401A	401B	401C	401D	401E	401F
4020	4021	4022	4023	4024	4025	4026	4027
4028	4029	402A	402B	402C	402D	402E	402F
4030	4031	4032	4033	4034	4035	4036	4037
4038	4039	403A	403B	403C	403D	403E	403F
4040	4041	4042	4043	4044	4045	4046	4047



The wall would only be solid if you ran into it from the left side of the screen. If you walked toward it from the right side, you would walk right through it. If, instead, you typed:

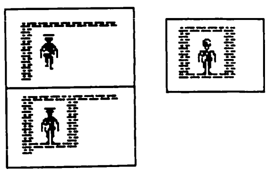
4013: 02
4014: 01
1209G

you would look like a fool no matter what side you hit the wall from.

Finally, I feel I should tell you that:

A) If you kill Hitler, you **can not** just teleport back to room 1 to win the game. The computer

Diagram 4



only registers a win if you go from room 2 to room 1. The "quickest way" is not always the best. Teleporting to room 2 and then walking down into room one on the other hand, does work.

B) Mr. Darrah writes that pressing Reset and then typing:

15AE: EA EA EA

will give you an infinite number of bullets. Technically, this is not true. All this does is stop the computer from subtracting bullets from your clip each time you fire, so you always have the same number of bullets. For this to work, you must have at least one bullet in your clip.

My compliments to Ray Darrah on a very informative as well as entertaining article on the deprotection of this game. Have fun.

Capturing the Hidden...

Archon Editor

by Steve and Rod Smith

Archon
Electronic Arts
2755 Campus Dr.
San Mateo, CA 944Ø3

Requirements:
A Deprotected copy of Archon
A sector editor
A 48K slave disk with no HELLO

Now that Steve McLendon has provided all of you with a way of deprotecting your Archons, we thought you might like to be in on a little secret program that Jim Nitchals has left inside his code. Yes, that's right! A secret program! To view this incredible discovery, boot up your Archon, and when you arrive at the options page, type in control-I, J, L, N, then any key. You should now be sitting in the Archon editor. A brief summary of the things you can do will follow, after we explain how to get it out as a file.

How We Discovered This Amazing Program

Well, it wasn't too hard. After having deprotected the program, we went snooping through the disk looking for strange text or leftover source code (a lot of programs are likely to have remnants or scrap data lying around). On track $11, sector $ØF we noticed text that said "NOP THE NEXT 3 BYTES FOR GOODIES". It didn't take long to figure out that meant track $11, sector $ØE, bytes Ø1-Ø3 because of the wrap around. There we put EA EA EA and booted the disk. To our surprise was an editor! Then we put a 4C 59 FF at the same spot and let it boot up. This allowed us to enter monitor and snoop around. That's where the secret key input sequence was discovered. So, our next concern was how to get the program into a single file, since it supported DOS, and none is loaded with Archon. Leaving in our 4C 59 FF at track $11, sector $ØE bytes $Ø1-Ø3, we set up for a boot by first clearing out memory with zeros by typing:

```
FF59G
3ØØ:ØØ
3Ø1<3ØØ.BFFEM
6 CTRL P
```

At the options page, press a button or something to start the game so that you will enter the monitor instead. Then we examine memory to see how much code was loaded. A quick look shows us that the program uses $803 up to $3FFF. To get DOS in memory, we will need to reboot. So, we have to move the $8ØØ page up to $4ØØØ to save it from being clobbered during a boot by typing:

```
4ØØØ<8ØØ.9ØØM
```

Now we boot up a normal DOS disk with no HELLO program so we don't wipe out anything else in memory. Then we move our $8ØØ page back down to where it belongs by typing (make sure you're in the monitor):

```
8ØØ<4ØØØ.4ØFFM
```

Before we save the entire file out, we must make a few changes in the code. First of course is to remove our 4C 59 FF. It is residing at $9Ø1 in memory and we need to replace it with EA EA EA. So we type:

```
9Ø1:EA EA EA
```

Now we would save the program out. However, hindsight tells us to make one more modification to the code. There are two parts to the Archon Editor. First is the shape editor and second is the hi-res page editor. We discovered a timing loop that will only allow you to remain in the hi-res editor for about ten minutes or so (very irritating!). So we will take out the counter by skipping over it before saving the final product. To do this we type:

```
3299:4C A9 32
```

Finally, we are ready to save our final file! Just type:

```
BSAVE EDITOR,A$8Ø3,L$38ØØ
```

and DOS will do the rest for you. Of course you may call the file anything you like. If you CATALOG your disk, you should see a 58 sector binary file under the name you saved it as.

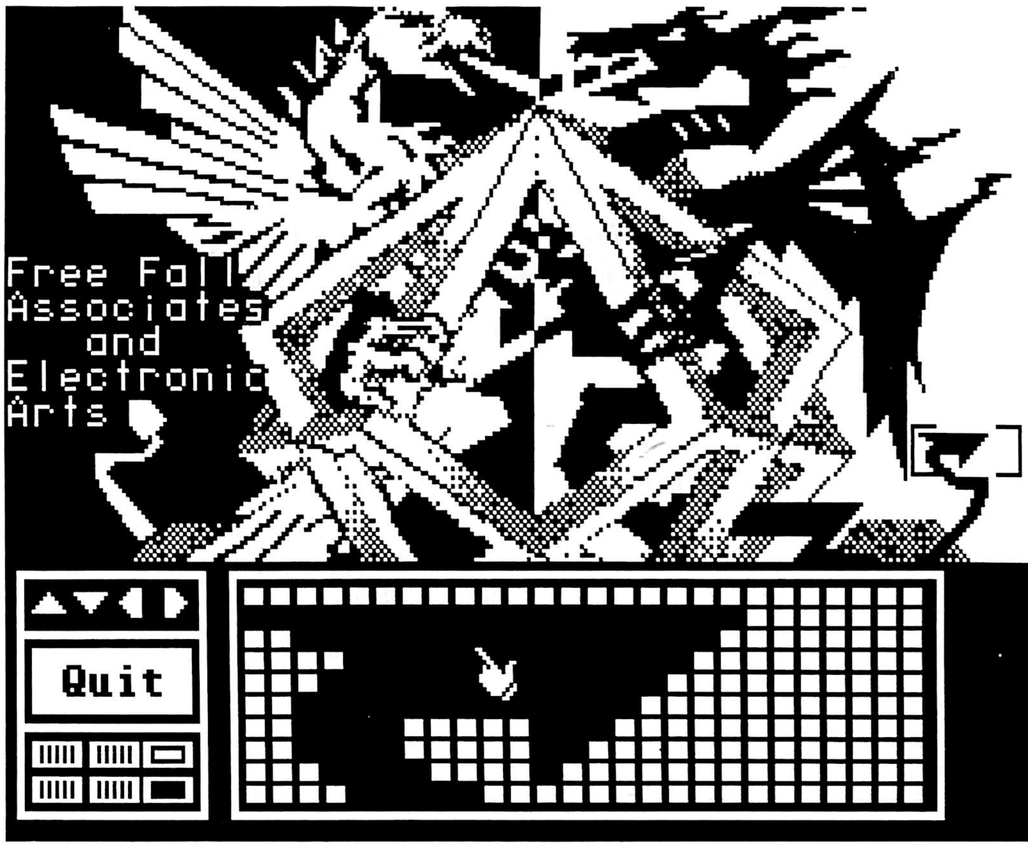
Some Instructions

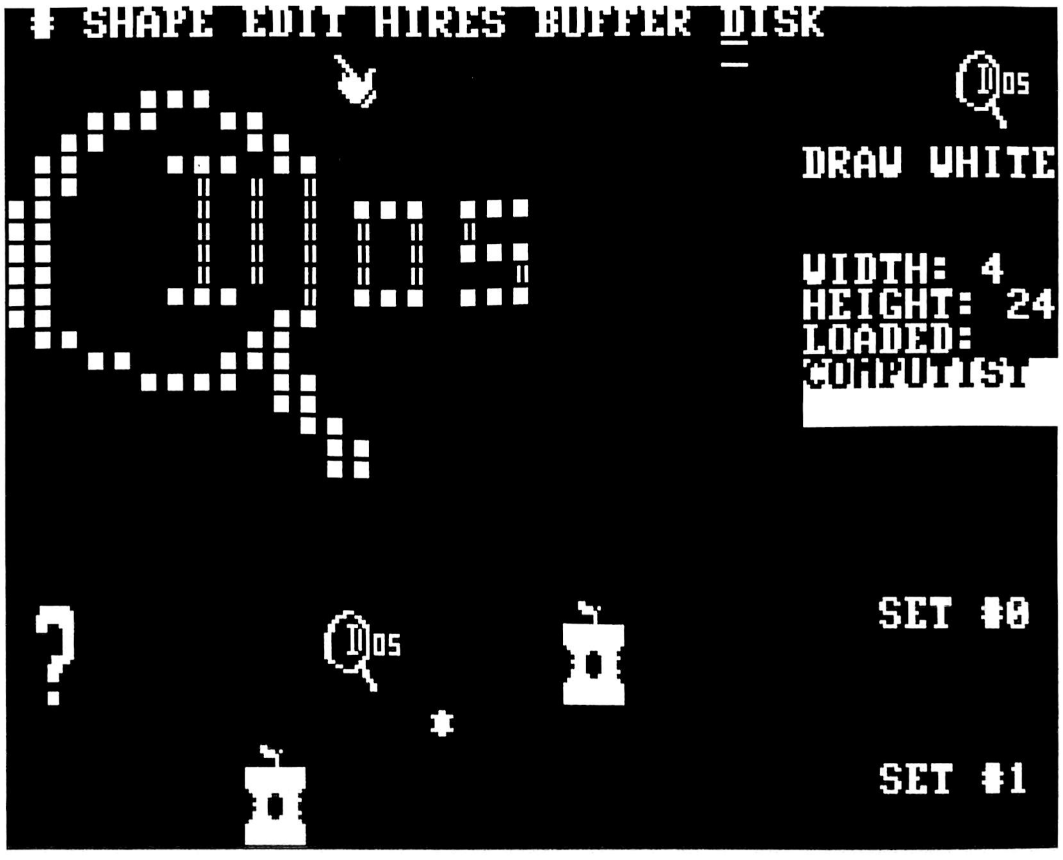
Following, is a summary of the things we discovered you may do with the editor. There are bugs in the DOS routines still but most everything is usable. A word about the DOS bugs will follow.

In addition to the commands listed in table 1, the program also supports the joystick. Just move the little hand around and press the button

Table 1: Editor Commands

CTRL-D	Step through DOS options
(Return) or CTRL-M :	Copy shape to spot indicated by *
CTRL-P	Select Plotting mode
CTRL-T	Sound click toggle
1	Flip bits in row on/off
4	Mirrors row
5,6	Scroll current row left or right
7	Change color set for row
8	Change color set for byte
#	Copy visible set to buffer
()	Select shape set numbers
*	Clears the editing area
<>	Increment/decrement width (1-4)
^	Increment height of shape (2-24)
(arrows)	Move asterisk
CTRL-W, CTRL-A, CTRL-S, CTRL-Z	Scroll editing screen
I,J,K,M	Move plotting cursor
(Space)	Plot/unplot pixel
2	Mirror shape
3	Vertical Mirror
R	Rotate shape 9Ø degrees
Y	Delete a row
B	Insert a row
G	Delete a column
H	Insert a column
!	Undo mistake
P	Get shape from set (at *)
&	Erase the buffer
N	Animate sequence
CTRL-C	Quit and reboot
E	Edit hi-res page (and return)
(=;*)	Erase hi-res page
W,A,S,Z	Move hand
I,J,K,M	Move edit window
(Space)	Plots/unplots/selects at hand
=	Another hi-res editing mode
P	capture highlighted block to shape
(Return)	Put shape at highlighted spot





at the selection. Note: the program, like Archon, requires 64K.

We have found that the saving of hi-res pictures option does not work properly. If you do modify a picture, and wish to save it, the Editor has placed it in bank #1 of the RAM card. To get it, you should exit the program and go into the monitor. Write enable the language card. Copy the F8 ROM to the language card, disable the ROM (select the RAM card) and save the picture, thusly:

```
C081 N C081
F800<F800.FFFFM
C083 N C083
2000<D000.F000M
```

Boot up DOS and save out your picture by typing:

```
BSAVE PICTURE,A$2000,L$1FFF
```

or whatever you wish to call it.

A Few Comments

This program is of high quality indeed. It is not known why Jim Nitchals chose to leave it in Archon as a secret program. However, after Archon][came out, we were anxious to see if our "secret" was still there. Indeed, it was. But it is not a fully working version. Instead it gives a P.O. Box as to where you may contact Jim Nitchals for the source code. Our (the authors') letter went unanswered.

With a little work, it should be simple to decode the shape files that get written to disk and use the shapes you create in your own programs. Have lots of fun with this apparent gift to hackers from Mr. Nitchals.

softkey for three from...

Mindscape

by Jeff Rivett

Mindscape Inc.
3444 Dundee Rd.
Northbrook, IL 60062

Requirements:
The Halley Project,
A View to a Kill
The Mist
48K Apple or compatible
A blank disk
Super IOB version 1.2 or 1.5

The Halley Project

The Halley Project: A Mission in our Solar System is a terrific new simulation-style game from Mindscape. It provides a very accurate model of our solar system in which you perform various piloting tasks, gradually building up your rank. Obviously, some compromises have to be made in order to stuff an entire solar system onto one side of a floppy disk, but those which have been chosen are completely acceptable, and don't detract from the overall playability of the game.

Unfortunately, the disk is copy protected. These days, I try not to waste too much time with bit copiers, so after trying a few copiers on default and having no success, I went to work.

The first thing I found is that the Address and Data Epilogue bytes on this disk have been changed to FF FF. I immediately wrote a controller for IOB to get rid of that obstacle. The copy booted up to the title page, where it appeared to do some kind of nibble-count.

I own a Replay II copy card and I get a lot of use out of it when I'm trying to deprotect disks. One of the nice things about the card is that it shows you the current program counter when you hit the button.

In this case, I waited until I was sure the nibble-count was happening, then hit the Replay button and took note of the PC. I then got into the monitor and looked for suspicious code in that area. Sure enough, I found a chunk of code that ended with a JuMP to C6ØØ, which of course boots the disk. A little farther back I noticed a JMP to 1DØØ. On a hunch, I decided to change the JMP to C6ØØ to a JMP to 1DØØ. First I had to find that piece of code on the disk, so I used the Tracer from CIA to search for the string 4CØØC6, and changed the code to 4CØØ1D. This time the disk booted perfectly.

Here's a Super IOB controller to do the whole thing. Just install it into Super IOB as usual and RUN it.

Halley Project controller

```
1ØØØ REM HALLEY PROJECT CONTROLLER
1Ø1Ø TK = Ø :LT = 35 :CD = WR :MB = 151
1Ø2Ø ST = Ø :T1 = TK : GOSUB 49Ø : RESTORE : GOSUB
     19Ø : GOSUB 21Ø : GOSUB 17Ø
1Ø3Ø GOSUB 43Ø : GOSUB 1ØØ :ST = ST + 1 : IF ST <
     16 THEN 1Ø3Ø
1Ø4Ø IF BF THEN 1Ø6Ø
1Ø5Ø ST = Ø :TK = TK + 1 : IF TK < LT THEN 1Ø3Ø
1Ø6Ø GOSUB 31Ø : GOSUB 23Ø :TK = T1 :ST = Ø : GOSUB
     49Ø
1Ø7Ø GOSUB 43Ø : GOSUB 1ØØ :ST = ST + 1 : IF ST <
     16 THEN 1Ø7Ø
1Ø8Ø ST = Ø :TK = TK + 1 : IF BF = Ø AND TK < LT THEN
     1Ø7Ø
1Ø9Ø IF TK < LT THEN 1Ø2Ø
11ØØ HOME :A$ = "ALL^ DONE" : GOSUB 45Ø : END
5ØØØ DATA 213 ,17Ø ,15Ø ,213 ,17Ø ,173 ,255 ,255
     ,255 ,255
5Ø1Ø DATA 1^ CHANGES
5Ø2Ø DATA 17 ,Ø ,154 ,29
```

controller checksums

1ØØØ	-	$356B	1Ø7Ø	-	$2BF4
1Ø1Ø	-	$85C6	1Ø8Ø	-	$5DC3
1Ø2Ø	-	$FC63	1Ø9Ø	-	$1A37
1Ø3Ø	-	$AD71	11ØØ	-	$ØC37
1Ø4Ø	-	$71B6	5ØØØ	-	$94BØ
1Ø5Ø	-	$B973	5Ø1Ø	-	$BF26
1Ø6Ø	-	$71EF	5Ø2Ø	-	$5EF6

The Mist & A View to a Kill

The Mist and **A View to a Kill** are two new Adventure games from Mindscape. They both run under the Runtime Pascal system from Apple. This is easy to see from some of the characteristics of the boot, such as a screen full of inverse 'at' signs (@), and a weird solid block cursor that appears and disappears at the top left of your screen.

How convenient it is when a company uses basically the same protection on all their disks. These disks both use the same altered Address and Data Epilogue bytes as The Halley Project. The nibble-count or whatever it is works a little differently, though, because it happens right on track Ø. Again, I searched for the JMP to reboot, but replaced it this time with an RTS ($6Ø).
Here's the controller for The Mist and A View to a Kill:

Mist / View to a Kill controller

```
1ØØØ REM MIST & VIEW TO KILL CONTROLLER
1Ø1Ø TK = Ø :LT = 35 :CD = WR :MB = 151
1Ø2Ø ST = Ø :T1 = TK : GOSUB 49Ø : RESTORE : GOSUB
     19Ø : GOSUB 21Ø : GOSUB 17Ø
1Ø3Ø GOSUB 43Ø : GOSUB 1ØØ :ST = ST + 1 : IF ST <
     16 THEN 1Ø3Ø
1Ø4Ø IF BF THEN 1Ø6Ø
1Ø5Ø ST = Ø :TK = TK + 1 : IF TK < LT THEN 1Ø3Ø
1Ø6Ø GOSUB 31Ø : GOSUB 23Ø :TK = T1 :ST = Ø : GOSUB
     49Ø
1Ø7Ø GOSUB 43Ø : GOSUB 1ØØ :ST = ST + 1 : IF ST <
     16 THEN 1Ø7Ø
1Ø8Ø ST = Ø :TK = TK + 1 : IF BF = Ø AND TK < LT THEN
     1Ø7Ø
1Ø9Ø IF TK < LT THEN 1Ø2Ø
11ØØ HOME :A$ = "ALL^ DONE" : GOSUB 45Ø : END
5ØØØ DATA 213 ,17Ø ,15Ø ,213 ,17Ø ,173 ,255 ,255
     ,255 ,255
5Ø1Ø DATA 1^ CHANGES
5Ø2Ø DATA Ø ,12 ,144 ,96
```

controller checksums

1ØØØ	-	$356B	1Ø7Ø	-	$2BF4
1Ø1Ø	-	$85C6	1Ø8Ø	-	$5DC3
1Ø2Ø	-	$FC63	1Ø9Ø	-	$1A37
1Ø3Ø	-	$AD71	11ØØ	-	$ØC37
1Ø4Ø	-	$71B6	5ØØØ	-	$94BØ
1Ø5Ø	-	$B973	5Ø1Ø	-	$BF26
1Ø6Ø	-	$71EF	5Ø2Ø	-	$5555

Softkey for Alphabetic Keyboarding

by Marc Lirette

South-Western Publishing Co.
5101 Madison Rd.
Cincinnati, OH 45227

Requirements:
Super IOB v1.5
Two blank disks or 2 sides of one disk

Alphabetic Keyboarding is a typing program that teaches you how to type. Like most other software on the market, this program is protected. Although the program can be copied using EDD, an unprotected copy is always nice to have.

The Protection

Upon receiving the program I examined the protection scheme used. I found that the DOS used was a fairly normal DOS although it was rather mixed up. By this I mean that the commands are moved around, the buffers are moved and its RWTS has been moved and mixed around.

I used TRAX from Bag Of Tricks and examined the disk. By using the raw track dump I was able to see that the third byte of the data header was changed from the normal AD to another number. This number varies from track to track and from sector to sector. Also the disk has an extra byte after the header and this can create problems if it is not copied properly. The controller following this softkey will copy the disk for you properly.

The Controller

In line 1005 of the controller the pokes will ignore reading errors, change the data header marker from AD to 96 and branch if it is not a 96. This will cause RWTS to read the data field if it is not the address field. The poke in line 1020 will change the location of the translate buffer by one byte so that it will ignore the extra byte. The poke in line 1030 puts buffer location back to normal before writing it back to the disk.

There are three parts to the controller. The first part will copy tracks 3 to 16 ($03 to $10 hex) which is protected. The second will copy track 17 ($11 hex) which is normal DOS 3.3 format. The third part will copy tracks 18 to 34 ($12 to $22 hex) which is also protected. The DOS is not copied since it is protected differently and you can put on your own fast DOS.

Note: When copying the first disk you should hear the drive moan and groan when it copies track $1E sectors 5, 6, and 7. This is because these three sectors are blank and the controller can't read them.

Procedure

1) Initialize your disk(s) with a fast DOS.

2) Install the controller into Super IOB v1.5 and run it.

3) Copy the Alphabetic Keyboarding disk. Do not use the format option.

4) Enjoy your backup.

controller

```
1000 REM ALPHA CONTROLLER
1005 POKE 47356 ,150 : POKE 47357 ,240
     : POKE 48712 ,24
1010 TK = 3 : LT = 17 : ST = 15 : LS = 15 : CD
     = WR : FAST = 1
1020 POKE 47349 ,87 : GOSUB 490 : GOSUB
     610
1025 POKE 47349 ,86 : GOSUB 490 : GOSUB
     610 : IF PEEK (TRK ) = LT THEN 1035
1030 TK = PEEK (TRK ) : ST = PEEK (SCT
     ) : GOTO 1020
1035 TK = 17 : LT = 18 : ST = 15 : LS = 15
1040 GOSUB 490 : GOSUB 610
1045 GOSUB 490 : GOSUB 610
1050 TK = 18 : LT = 35 : ST = 15 : LS = 15
1055 POKE 47349 ,87 : GOSUB 490 : GOSUB
     610
1060 POKE 47349 ,86 : GOSUB 490 : GOSUB
     610 : IF PEEK (TRK ) = LT THEN 1070
1065 TK = PEEK (TRK ) : ST = PEEK (SCT
     ) : GOTO 1055
1070 HOME : PRINT "COPYDONE" : END
```

Softkey for...

HACKER

by Jeff Rivett

Activision
P.O. Box 7287
Mountain View, CA 94039

Requirements:
48K Apple][or clone
A blank disk
A disk copy program

It was surprising to find that this disk is protected. I thought that Activision would figure that it was a waste of time to protect a program aimed at hackers. Wrong.

The first thing I usually do when I get a new program is to check which tracks, if any, are formatted in some non-standard way. The easiest way to do this is to try and read it with Locksmith's fast disk backup. HACKER read perfectly, so I immediately made a copy of the disk.

On booting the copy, I found it worked ok, except that I noticed the drive recalibrated (returned to track zero) and stayed there for a while at one point during the boot. This made me a little suspicious, so I decided to get into the game a little just to make sure nothing went screwy. Sure enough, after I had progressed past the first few stages of the game, it hung up completely.

I rebooted the disk to check on that recalibration at this point. This time I also noticed the Applesoft prompt at the bottom of the screen during the boot, and that it disappeared just after the recalibration. The prompt isn't as promising a sight as it used to be when trying to deprotect disks, but it sometimes gives you an indication of a 'near DOS 3.3' type of DOS.

For a laugh, I tried booting the disk again, but this time I hit CTRL C as soon as I turned the power on. This would normally cause an Applesoft boot program to break. Surprise, surprise! It worked. The boot program was there, and listable. I tried CATALOG, which also worked.

The boot program BLOADs a large binary file, then BRUNs a smaller file, called HACKB. To confirm a hunch, I typed MONICO to turn on the DOS monitor, and then RUN. The first file loaded, then the HACKB started to run. Almost immediately, the drive recalibrated.

At this point I printed out a Disassembled listing of HACKB. It loads at $5E00, and its length is $1E1. This chunk of code does a lot of weird stuff, but all I was interested in was disk access. Disk access code looks like LDA $C089,X which turns on the drive motor. Almost anything that accesses memory in the range $C080,X to $C08F,X is a command to the drive controller card. Other code to look for is any call to the RWTS, which is $B7B5 in normal DOS.

I found two subroutines in HACKB which mess with the disk. One is at $5E5D - $5E6C, and the other at $5E6D - $5EC2. The first routine just sets up for and calls the RWTS to send the drive head back to track zero, which is the recalibration I noticed. The other one is a little more complicated, but all it really does is look for some special bytes on track zero.

What I then looked for was any code in the second routine which alters memory, like a STA or a STX, which store stuff in memory. I found two pieces of code like this, at $5EA3, which stores the Y register in $FC, and at $5EB4, which stores the Accumulator in $5E18.

The code in this routine is fairly linear, in that it doesn't do any jumping out of itself, and the looping it does is usually back a few steps only. The trick here is to find out what $FC and $5E18 are supposed to have in them after the routine checks track zero. So I put a JMP to $FF59 at $5EB7, so that after the track zero check, and after the two special locations are set, we jump to the monitor. I then ran the program, and when the monitor prompt appeared, I checked $FC and $5E18 to see what they came up with. They were, respectively, $FF and $55.

Now all I did was replace the code that checked track zero with a couple of lines of code to set the two locations, and saved HACKB. The changes are listed below. The new version did not hang. I have been all the way through the program, and it worked just fine.

The changes made to HACKB:

```
5E76-   A9 FF        LDA   #$FF
5E78-   85 FC        STA   $FC
5E7A-   A9 55        LDA   #$55
5E7C-   4C B4 5E     JMP   $5EB4
```

Step by step:

1) Make a copy of the disk with COPYA or a similar copier.

2) Boot the copy and hit CTRL C quickly to break out of HELLO.

3) Fix HACKB to work on the copy.

BLOAD HACKB
CALL -151
5E76:A9 FF 85 FC A9 55 4C B4 5E
BSAVE HACKB,A$5E00,L$1E1

That's it! Hack away.

softkey for...

Disk Director

by A. L. Head, Jr.

Sensible Software
24Ø11 Seneca
Oak Park, MI 48237

Requirements:
48K Apple][and up
FID from the DOS 3.3 System Master
Disk Director
Two blank disks
Super IOB v1.5 (Optional)

Disk Director, produced by Sensible Software and licensed to Softsmith for distribution, is a great disk utility for unprotected disks. There are three major programs on the disk as follows: Disk Organizer, Disk Recovery and Super Disk Copy. The Disk Organizer permits one to alphabetize the files on a disk, change the HELLO program, delete and undelete files, add titles within the catalog, lock and unlock files, Zap DOS, show the track/sector map, et cetera. The Disk Recovery checks for bad sectors, locks them out, rebuilds the VTOC, lists the files with bad sectors, et cetera. Super Disk Copy features a Quick Copy that only copies sectors that are used; a Contiguous Copy that re-arranges the track/sector usage of the files in a continuous manner; and a Brute Force Copy that copies a disk sector by sector. DOS or the entire disk can be copied. Control characters are highlighted in the catalog. A very useful feature is the track/sector map.

The Protection

On to the examination of the disk. This disk has a modified DOS on it. Tracks $ØØ through $Ø2 use address and data epilogues of $FF FF instead of the standard $DE AA. In addition, the address field checksums are set to various values. Tracks $Ø3 through $22 are in standard DOS 3.3 format except that the address fields checksums are non-zero. Tracks $Ø3 through $1Ø are empty. A softkey can be determined with the above information.

The Softkeys

This disk is easily deprotected. The modified DOS will be circumvented by installing normal DOS on the copy. I will give you two different ways of doing it here. The first way will use the FID program on the System Master. If you have Super IOB 1.5, the second way is easier. I have written a controller to normalize the disk.

I will start with the FID method.

1) Boot the DOS 3.3 Master disk.

2) Insert a blank disk and initialize it with no HELLO program:

```
FP
INIT HELLO
DELETE HELLO
```

3) Apply a patch to DOS in memory to ignore address checksums:

```
CALL -151
B98A:ØØ
```

4) Get FID running and copy the Disk Director.

```
BRUN FID
```

When the FID display appears, use the wildcard (=) to copy every file to the duplicate disk. Do not select ''Prompting'.

5) When the copying is complete, boot the copy. When the BASIC prompt appears (probably with a FILE NOT FOUND), type NEW and then type in the program given below. Alter the filename to match the original HELLO program on the disk if necessary. Note that the filename is the full thirty characters allowed by DOS.

```
1Ø PRINT CHR$ (4 ) "BRUN^HELLO^^^^^^^^^^
     ^^^^^^^12/19^14"
```

And save the program, of course.

```
SAVE HELLO
```

We're pretty much done at this point. If you wish, you may pack the files on the disk closer together by copying everything to another blank disk using the Contiguous Copy option on the Disk Director. Then you may re-use the first backup for something else.

You now have a completely normalized copy of Disk Director.

The Super IOB Method

1) Initialize a blank disk the same way as in steps 1 and 2 above.

2) Install the Controller with this article into Super IOB 1.5 and start Super IOB.

3) Copy the Disk Director disk onto your newly initialized disk. Do not use Super IOB's Format option.

The controller given at the end of this article begins copying the original at track $11. All files are contained on the original disk between tracks $11 and $22.

4) Now all you need to do is add the HELLO program to the disk as in step 5 of the FID method.

Et Cetera...

The copy of Disk Director produced here retains all the features of the original except the Softsmith logo which is contained in one of the boot stages. A more enterprising person might search for it. I will leave that to others.

Many other software products use a variation of the protection used on this disk; that is, a modified DOS with the files written to disk in more or less standard form. When this is the case, standard DOS can be written to the copy by initializing the disk as done above. The altered marks and/or checksums can be taken care of by patching DOS and then using FID or a similar program to transfer the files from the original to the copy. Then a HELLO program can be written to run the program or whatever as done here. Using Super IOB 1.5 provides an easier way to accomplish the same end. In virtually all such cases the use of a fast DOS (I use Pronto-DOS with the enhancement to print the number of free sectors in the CATALOG) instead of DOS 3.3 on the copy disk will speed up the running and loading of files by a factor of about three. Disk Director provides an option to copy just the DOS portion of a disk to another disk. I have never encountered any difficulty by doing this. A disk done this way is much faster than the original.

Any of the popular bit-copy programs will copy DISK DIRECTOR with their default settings and produce a protected copy. I have used COPY][PLUS and E.D.D. III.

controller

```
1ØØØ REM DISK DIRECTOR CONTROLLER
1Ø1Ø TK = 17 :LT = 35 :ST = 15 :LS = 15 :CD = WR
     :FAST = 1
1Ø2Ø GOSUB 49Ø : GOSUB 27Ø : GOSUB 61Ø
1Ø3Ø GOSUB 49Ø : GOSUB 61Ø : IF PEEK (TRK ) = LT
     THEN 1Ø5Ø
1Ø4Ø TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO 1Ø2Ø
1Ø5Ø HOME : PRINT "DONE^WITH^COPY" : END
```

controller checksums

```
1ØØØ  - $356B     1Ø3Ø  - $26EØ
1Ø1Ø  - $1CD4     1Ø4Ø  - $A94E
1Ø2Ø  - $9284     1Ø5Ø  - $4914
```

Revisiting

LODE

by Steve Marvin

Lode Runner
Broderbund Software
1938 Fourth Street
San Rafael, CA 94901

Requirements:
At least 48K
DOS 3.3 or equivalent
Super IOB v1.5
Lode Runner
A blank disk

Tom Phelps' softkey for Lode Runner by Broderbund Software (Hardcore COMPUTIST No. 22), creates a 114 sector program file and requires a method to reset into the monitor. It also writes code over the visible text page on boot and preserves Lode Runner's reset vectors, which force a reboot on Reset from the game.

Although I have a modified CD ROM (Synergistics) in my //e which allows a cold boot to the monitor, with Lode Runner I ended up with a memory full of zeros. Lode Runner copies the monitor to the language card and alters the reset pointer bytes $FFFC-FFFD to $200, a subroutine which wipes RAM and re-boots.

For these reasons I had to do some extra work for my backup copy and will share this with others in the same predicament. If you were successful with Mr. Phelps' method but wish to trim the program, just enter the LR.MOVES hexdump and proceed to step 6.

My Method

The infamous boot code trace, a six step process involving monitor code moves, studying dumps and stack tracing led to a binary file LR.LOADER, the creation of which is discussed later in the article. To quickly sum it up, Boot1 (the first sector on track 0) decodes two subroutines and four addresses (by Exclusive-ORing with $A5) and puts them in the zero page and the stack. It then performs some jumps by pretending that the addresses in the stack were left there by JSRs and "returning" to them with RTSs.

The subroutines it performs are as follows:

$00B4: sets up header bytes of $D4 D5 D6 and a trailer byte of $D7.

$0060: finds and reads 4 sectors with the above header and trailer (No address field, the data is 4+4 encoded, with no address field) into $400 through $7FF.

$400: No action, just an RTS ($60).

$401: Changes the header bytes to $DD $F5 $D4, with a trailer of $D4, and steps by phases past track $0D. Then it reads one 4+4 encoded sector over the code at $400, turns the drive off, and jumps to $400.

The new code at $400, together with the previous boot code in $500-$7FF, moves inward continually by specific numbers of phases and reads all the remaining code that Mr. Phelps found after his reset. This includes a normal DOS 3.3 RWTS in $B600-$BFFF (except for an altered JuMP in the INIT handler). Then a short reset handler is placed in $200 (the input buffer) and all reset pointers are changed to point to it. The RAM card copying (and altering) is also done at this time.

By investigation, comparison and experimentation I found that (a) a Normal RWTS can be used, (b) the code from $400-$8FF is never reused after booting, (c) the code and vectors in pages two and three can be altered without damage, (d) a short series of pokes takes care of page 0, (e) the stack is reinitialized each reset, (f) there is no code or data loaded at $1F00-$1FFF, and (g) $8F00-$8FFF is zeroed on boot.

Using this information I produced a 93 sector BRUN program which does not display garbage while booting, and will reset to the beginning of the game rather than re-boot (A nice feature, I think).

The Procedure

1) INIT a DOS Slave disk with a binary Hello program called LODE RUNNER.

```
NEW
CALL-151
9E42:34
3D3G
INIT LODE RUNNER
DELETE LODE RUNNER
```

Note: You may use any DOS with a normal RWTS as long as $BE0E holds the INIT handler's address and the INIT handler lives at $BEAF.

2) Install the IOB controller at the end of this article and run Super IOB to capture tracks 3 through $C onto your newly initialized disk. (This controller uses Super IOB v1.5's R/W a Range Quickly routine and performs sector edits to correct the VTOC for track 3-$C usage).

3) Build the binary file LR.MOVES (Mr.

RUNNER

Phelp's subroutines with reset and setup code added).

CALL -151

```
1FØØ: AØ 4F A9 9B A2 1C 2Ø 6Ø   $5DA8
1FØ8: 1F AØ 2Ø A9 6Ø A2 2F 2Ø   $4EEF
1F1Ø: 6Ø 1F A9 4Ø 85 E6 2Ø F2   $61AC
1F18: F3 AØ 4Ø A9 Ø9 A2 Ø6 2Ø   $FE47
1F2Ø: 6Ø 1F AØ 4Ø A9 8F A2 ØC   $87B2
1F28: 2Ø 6Ø 1F AØ ØØ 8C ØE BE   $ØF56
1F3Ø: 8C CC BE A9 8E 8D ØF BE   $7296
1F38: A6 2B 8E FF Ø2 AØ 48 B9   $C4C8
1F4Ø: 7B 1F 99 ØØ Ø2 88 1Ø F7   $3CBE
1F48: A9 Ø2 AØ ØØ 8C F2 Ø3 8D   $C496

1F5Ø: F3 Ø3 8C FØ Ø3 8D F1 Ø3   $9B21
1F58: 49 A5 8D F4 Ø3 4C 19 Ø2   $FFBØ
1F6Ø: 84 Ø1 85 Ø3 A9 ØØ 85 ØØ   $8FFE
1F68: 85 Ø2 AØ ØØ B1 ØØ 91 Ø2   $D5EB
1F7Ø: C8 DØ F9 E6 Ø1 E6 Ø3 CA   $24CD
1F78: DØ FØ 6Ø A9 D2 2C A9 DØ   $3A4E
1F8Ø: 2C A9 CC 2C A9 A1 48 2Ø   $8AD5
1F88: 2F FB 2Ø 58 FC 2Ø 84 FE   $46FC
1F9Ø: 68 8D ØØ Ø4 A9 Ø6 85 8C   $57FB
1F98: A9 FF 85 99 A9 CA 85 95   $56C9

1FAØ: A9 4C 85 23 A9 5Ø 85 36   $Ø7A1
1FA8: A9 8E 85 37 A9 B5 85 38   $49C5
1FBØ: A9 B7 85 39 4C ØØ 6Ø      $A84F
```

3D3G
BSAVE LR.MOVES,A$1FØØ,L$ØB7

4) Create the file LR.LOADER as follows:

MAXFILES 1
CALL -151
96ØØ<C6ØØ.C6FFM

```
9564: 4C ØØ 96 A9               $ØC13
9568: ØØ 8D Ø6 Ø1 A9 Ø4 8D Ø7   $FC28
957Ø: Ø1 A9 83 8D 75 Ø4 A9 95   $7E72
9578: 8D 76 Ø4 A9 Ø9 8D 6E Ø4   $A1D5
958Ø: 4C Ø1 Ø4 AØ ØØ B9 ØØ Ø9   $Ø819
9588: 99 ØØ Ø4 88 DØ F7 A9 9B   $A6F9
959Ø: 8D 71 Ø4 A9 95 8D 72 Ø4   $Ø893
9598: 4C ØØ Ø4 9D 88 CØ A2 Ø9   $2DDF
95AØ: AØ ØØ B9 ØØ ØØ 99 ØØ 2Ø   $962C
95A8: 88 DØ F7 EE A4 95 EE A7   $3611

95BØ: 95 CA DØ EC 4C 59 FF A9   $6B7E
95B8: 66 8D Ø6 Ø1 A9 95 8D Ø7   $2DBF
95CØ: Ø1 AØ 47 B9 ØØ Ø9 99 ØØ   $3A56
95C8: Ø8 88 DØ F7 6Ø A9 1D 8D   $D99Ø
95DØ: 59 96 AØ ØØ B9 ØØ Ø8 99   $DC7B
95D8: ØØ Ø9 88 DØ F7 A9 4C 8D   $AØFC
95EØ: 45 Ø8 A9 B7 8D 46 Ø8 A9   $C45D
95E8: 95 8D 47 Ø8 A9 EA AØ ØB   $39DE
95FØ: 99 16 Ø8 88 DØ FA A9 Ø1   $8EE8
95F8: 8D F9 96 A9 Ø8 8D FA 96   $E1DØ
```

96F8:4C CD 95
BSAVE LR.LOADER,A$9564,L$197

5) Insert your original Lode Runner disk in the drive and type

9564G

The drive will recalibrate twice, the screen will fill with garbage, and you will hear the lengthy Lode Runner boot ending with a bell and the monitor prompt. The drive will turn off at this point. (A disassembly of the loader will reveal what happened.)

6) Compact the code and insert the entry jump:

2ØØØ<6ØØØ.8EFFM
4FØØ<9BØØ.B5FFM
ØEFD:4C ØØ 1F

7) Boot your DOS Slave with LR.MOVES on it:

6 CTRL P
BLOAD LR.MOVES,A$1FØØ

Change to your new Lode Runner Disk.

BSAVE LODE RUNNER,A$ØEFD,L$5BØ3

That's it !!

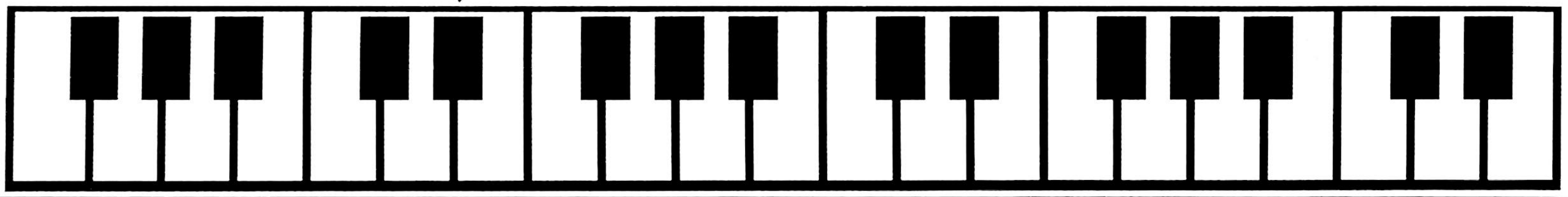
controller

```
1ØØØ REM LODERUNNER SCREEN CONTROLLER
1Ø1Ø TK = 3 :LT = 13 :ST = 15 :LS = 15 :CD = WR :FAST
     = 1
1Ø2Ø T1 = TK : GOSUB 49Ø : RESTORE : GOSUB 17Ø :
     GOSUB 27Ø : GOSUB 61Ø
1Ø3Ø GOSUB 23Ø : GOSUB 49Ø : GOSUB 61Ø : IF
     PEEK(TRK ) = LT THEN 1Ø5Ø
1Ø4Ø TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO 1Ø2Ø
1Ø5Ø CD = RD :TK = 17 :LT = 17 :ST = 15 :LS = 15
1Ø6Ø GOSUB 5ØØ : GOSUB 61Ø
1Ø7Ø A$ = "3644:ØØ^ N^ 3645<3644.3669M^ N^
     27Ø3:ØØ^ N^ 27Ø4<27Ø3.27FEM^ N^ D9C6G"
1Ø8Ø FOR A = 1 TO LEN (A$ ) : POKE 511 + A , ASC (
     MID$ (A$ ,A ,1 ) ) + 128 : NEXT : POKE 72 ,Ø
     : CALL - 144
1Ø9Ø GOSUB 49Ø : GOSUB 61Ø
11ØØ HOME :A$ = "DONE^WITH^COPY" : GOSUB 45Ø :
     END
5ØØØ DATA 222 ,235 ,222 ,17Ø
```

controller checksums

```
1ØØØ  - $356B      1Ø6Ø  - $6Ø46
1Ø1Ø  - $1336      1Ø7Ø  - $953A
1Ø2Ø  - $4C8D      1Ø8Ø  - $2E9Ø
1Ø3Ø  - $7914      1Ø9Ø  - $7196
1Ø4Ø  - $35F5      11ØØ  - $A963
1Ø5Ø  - $9668      5ØØØ  - $899Ø
```

MIDI/4



by Ray Darrah

MIDI/4
Passport Designs Inc
785 Main St.
Half Moon Bay, CA 94Ø19

Requirements
48K Apple][Plus or greater
MIDI interface

It really upsets my stomach when a company protects a piece of software that requires hardware to run. Such is the case with MIDI/4, a decent multi-track recording package. I am still trying to figure out why they protected this software since it requires a MIDI interface before it will operate. Is Passport Design afraid users will pirate their hardware as well as their software?

As usual, my strong dislike for protected programs (especially those of a utility nature) coupled with Passport's audacity has compelled me to deprotect the little sucker. In addition, I find that after I deprotect a program, I sleep better at night.

An Overview

At first, I tried copying the disk with my favorite whole disk copier: Locksmith Fast Copy. This showed me that the disk was normal DOS except for tracks $Ø3 and $1Ø. Of course, the copied disk would not boot.

I then examined tracks $Ø3 and $1Ø to discover that these tracks were of a nibble count nature rather than an altered DOS nature. As a matter of fact, listening to the boot I could tell that the nibble count was most probably occurring right after the title page was loaded in. The next step was then to locate the nasty nibble count routine and defeat it.

Using my trusty ABACUS Know-Drive, I interrupted the program during the dirty nibble count and discovered that it lived somewhere above $8DØØ and below $95F9. Watching the disk head during the boot told me that it only nibble counted track $1Ø. But if that one failed, the head would move to track $Ø3. Perhaps track $Ø3 is an exact copy of track $1Ø (sort of a backup copy of the nibble counted track).

Searching the disk for the nibble count code revealed that it was contained in a file called READER. I spent a few hours tracing the code in the READER file and decided that this was not the way. I congratulate whoever wrote this code for creating another Frankenstein.

I then scrutinized the boot file sequence by putting a few MONCIO's here and there and discovered that the first part of it went like this: First, the Diversi-DOS on the disk loads and executes HELLO which executes DDM, loads the title picture (PDIM4P1) and executes M4XX. The binary file M4XX then loads READER and executes it via the JMP $8DØØ at $958. If everything goes O.K. in the READER file (nibble counts etc.), BRUNM4P is executed and the program magically comes to life!

With this knowledge in hand (or mind), I decided to re-route the boot sequence to eliminate execution of that bugger known as the READER file. My new boot sequence then would be like this: First, the Diversi-DOS on the disk loads and executes HELLO which executes DDM, loads the title picture (PDIM4P1), loads BRUNM4P and executes a modified M4XX. The modified M4XX file then loads READER and executes BRUNM4P via the modification.

This modifed boot sequence almost worked except it kept bombing into the monitor at $AØ4. I noticed at this point a $AØAG would start up the program. So I searched the BRUNM4P file for a JMP to somewhere in $AØØ. I changed the JMP $AØ1 at location $8Ø69 to a JMP $AØA and, voila! MIDI/4 was now broken!

Exact details of the deprotection now follow.

Step by Step

1) Copy your original MIDI/4 disk with some whole disk copier that can ignore errors on tracks $Ø3 and $1Ø (like Locksmith Fast Copy).

2) Put your original MIDI/4 disk away and LOAD HELLO from the copy.

3) Change line 6 of HELLO to read:

```
6 PRINT CHR$ (4) "BLOAD^ BRUNM4P" : PRINT CHR$
    (4) "BRUNM4XX"
```

4) Resave HELLO to the copy.

```
UNLOCK HELLO
SAVE HELLO
LOCK HELLO
```

5) Modify the M4XX program to execute BRUNM4P instead of READER.

```
FP
BLOAD M4XX
CALL -151
958:4C ØØ 8Ø
UNLOCK M4XX
BSAVE M4XX,A2Ø49,L346
LOCK M4XX
```

5) Modify the BRUNM4P program to JuMP to $AØA instead of $AØ1.

```
BLOAD BRUNM4P
8Ø69:4C ØA ØA
UNLOCK BRUNM4P
BSAVE BRUNM4P,A32768,L636
LOCK BRUNM4P
```

That's it! Enjoy.

Many thanks to Raymond Feruski.

We are NOT

PIRATES!

but we're not fools, either.

We're serious programmers and software users who just want to have backup copies of any software we own. COMPUTIST magazine shows us HOW TO MAKE BACKUPS OF COMMERCIAL SOFTWARE regardless of the maker's attempt to stop us from having legal copies. Don't let them stop you from protecting your own rights.

Remove copy-protection

from your valuable library of expensive software. The publisher of COMPUTIST has been showing subscribers how to unlock and modify commercial software for the past 4 years. Don't be one of the users abused by user-FIENDLY locked-up software. Subscribe.

6 issue SUBSCRIPTION RATES:

U.S.: $20
U.S. First Class: $24
Canada, Mexico: $34
Foreign Air: $60

SAMPLE COPY:

U.S.: $4.75
Foreign: $5.70

☐ NEW subscriber ☐ Renew my subscription

Name____________________ ID#________

Address ____________________

City ________ State ____ Zip ______

Country ________ Phone ________

VISA MasterCard ____-____-____-____ Exp. ____

Signature ____________________ CP28

US funds drawn on U.S. bank. In Washington add 7.8% tax. Send check or money order to:

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

5 FREE DISKS

With every set of 20 disks you order
You can get sets for as low as

$17.25

That's a total of 25 disks for as little as

69¢ a disk

SS/DD Namebrand
5¼" floppies
100% guaranteed
& include:
reinforced hubs
& write-protect tabs

Send me _____ sets without sleeves at $17.25 per set
Send me _____ sets, with Tyvek sleeves at $18.50 per set.
Add $3 shipping and handling for the first set, $1 for each additional set. Foreign orders add 20%. U.S. funds drawn on U.S. bank. Washington orders add 7.8% sales tax. Send your orders to:

Softkey Publishing; PO Box 110846-T; Tacoma, WA 98411

Name ____________________ ID# ________

Address ____________________

City ________ State ____ Zip ______

Country ________ Phone ________

VISA MasterCard ____-____-____-____ Exp. ____

Signature ____________________ CP28

A nibble of history...

Creative Computing bytes the dust

In memory of the demise of *Creative Computing*, our old adversary (one of many), we present a nibble of history from the heydays of Hardcore Computing...

Creative Computing Censors Hardcore Advertisement

Creative Computing is still censoring ads and suppressing information about copy protection. It has just refused to publish the HARDCORE advertisement shown on this page.

What did the editors of CC find unprintable in our ad?

"Only two lines," explained their Advertising Manager, Jerry Thompson. "They are: 'back up any diskette' and 'do & undo copy-protection'."

Jerry said that either we agree to let them remove those lines or they would refuse to run the ad. Well, we stood by our editorial policy of fighting censorship and they stood by their policy of information suppression. So you won't see this ad in CC.

One of their editors, George Blank, defended CC's policy by offering the lame comparison to Reader's Digest's policy against running liquor ads. He added that at least CC will run ads for competitors.

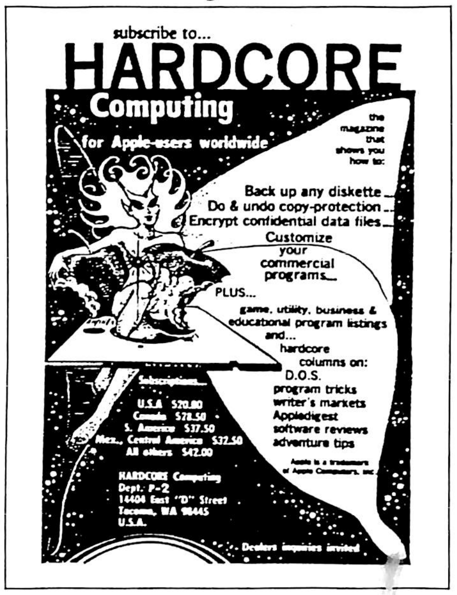
Big Deal! All of them share CC's censorship policy. All of them, except **Hardcore Computing**. And they will not run our ads.

George's final defense was that CC had a readership of well over 100,000... so obviously they must be doing things right. What can I say to that?

And when I mentioned that the readers have had no alternative, he answered promptly, "Yes, they do!" (meaning his competitors). I retorted that they practice the same form of censorship.

"That might mean that we're right and you're wrong." he concluded.

So...The battle goes on.



CC is probably just the first computer magazine to censor HARDCORE's ads. We plan to try to put one in *Call-A.P.P.L.E.*, but unless Val Golding, the editor, has altered his policy, they will probably refuse to run our ad, too.

We'll keep you posted on further developments.

Addendum: *Call- A.P.P.L.E.* didn't run our ad, but we learned from Grawin Publications that it wasn't Val Golding who had our ad pulled from its pages at the last moment. It was the doing of Dick Hubert, President of *Call A.P.P.L.E.*, and its Executive Director.

George Blank's reply...

Dear editor,

You quoted me correctly and for that I am grateful. I do not feel that you quoted me in context. For example, my "lame excuse" that Readers's Digest refused liquor ads came at a point in our conversation when we were discussing the fact that the courts have allowed publishers the right to set their own standards for accepting advertising.

I invite you to practice what you preach. If software prices are so extortionate, how do you justify $20 a year for your few miserable pages of hard-to-read mimeographed text? For the same price, your readers could buy over 2500 pages of Apple material printed in Creative Computing, or even over 400 pages of Apple material if you want to throw away all our ads, general articles, and material on other computers!

The truth is most current software companies are not profitable. We know; we keep getting bankruptcy notices instead of payment for advertisements, even from established firms. And it is casual copying, not professional pirates, that hurts profits most. In my own case, I no longer write programs for publication; it is more profitable to write custom software.

Sincerely yours,
George Blank,
Editor, Creative Computing

The text above was reprinted from an editorial in Hardcore Update 1.1, September 1981 which followed the printing of the Premier Issue.

Back Issues

of COMPUTIST (formerly Hardcore Computist)

are still available.
Some issues (marked NA) are sold out,
but library disks are available for all issues of COMPUTIST
and even the old COREs.

Don't

T Y P E

in programs that appear in COMPUTIST.

Order the Library Disk, instead!

Each month a Library Disk with all the programs that appeared in the previous issue of COMPUTIST is prepared for **SMART READERS** like you who have *better* things to do with their time than type in program listings. Please use the order form to order both magazines and library disks.

Back Issues and Library Disk order form

Issue	Magazine $4.75	Disk $9.95	Issue	Magazine $4.75	Disk $9.95	Issue	Magazine $4.75	Disk $9.95	Issue	Magazine $4.75	Disk $9.95
28	☐	☐	19	☐	☐	10	NA	☐	1	☐	☐
27	☐	☐	18	☐	☐	9	NA	☐			
26	☐	☐	17	☐	☐	8	NA	☐	Core 3	☐	☐
25	☐	☐	16	☐	☐	7	☐	☐	Core 2	☐	☐
24	☐	☐	15	NA	☐	6	NA	☐	Core 1	☐	☐
23	☐	☐	14	NA	☐	5	NA	☐			
22	☐	☐	13	☐	☐	4	☐	☐			
21	☐	☐	12	NA	☐	3	NA	☐			
20	☐	☐	11	NA	☐	2	NA	☐			

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

Name ______________________________ ID# ________

Address ______________________________

City ______________ State ________ Zip ____________

Country ______________ Phone ______________

VISA MasterCard ________-________-________ Exp. ________

Signature ______________________________ CP28

Send check or money order to:

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

Most orders shipped UPS.
Please use street address.
In Washington state:
add 7.8% sales tax.
Foreign orders:
add 20% shipping and handling.
US funds drawn on US bank.

Offer good while supply lasts.

Description of Available Back Issues

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

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 | *Features* | 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 | *Features* | 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 | *Features* | Essential Data Duplicator copy parms | *Core* | Direct Sector Access From DOS................

23 *Softkeys* | Choplifter | Mufplot | Flashcalc | Karateka | Newsroom | E-Z Draw | *Readers' Softkeys* | Gato | Dino Eggs | Pinball Construction Set | TAC | The Print Shop: Graphics Library | Death In The Caribbean | *Features* | Using A.R.D. To Softkey Mars Cars | How To Be The Writemaster | *Core* | Wheel Of Money | ....

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 | .....

21 *Softkeys* | DB Master version 4+ | Dazzle Draw | Archon | Twerps | *Readers' Softkeys* | Advanced Blackjack | Megaworks | Summer Games | College Entrance Exam Prep | Applewriter revisited | *Features* | Demystifying The Quarter Track | *Core* | Proshadow: A ProDOS Disk Monitor.............................

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 | *Feature* | 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 |

18 *Softkeys* | Scholastic Version of Bank Street Writer | Applewriter //e | SSI's Non-RDOS Disks | *Readers' Softkeys* | BPI Accounting Programs and DesignWare Programs | *Features* | Installing a Free Sector Patch Into Applewriter //e | Simple Copy Protection | *Core* | The Games of 1984: In Review | 65C02 Chips Now Available | Checksoft v2 | ....

17 *Softkeys* | The Print Shop | Crossword Magic | The Standing Stones | Beer Run | Skyfox | and 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 Part Two..

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

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

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 | and 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 | ....

Back issues not listed are no longer available.

But disks are still available for ALL sold- out issues !

Use the order form on the other side of this page

Writer's Guide

COMPUTIST

is a monthly magazine dedicated to the serious user of the Apple (or compatible) computer. COMPUTIST welcomes articles on a variety of subjects in all levels of technical difficulty but requires accurate data, technical competence, correct English usage, readable style, and fully defined jargon and buzzwords.

MANUSCRIPT MECHANICS

All manuscripts must be typed or printed on one side of the paper. Text should be double-spaced.

Printouts should use a non-compressed font with both upper and lower case. A letter quality mode is preferred, with each page torn at the perforation only. Pages need not be stapled together. The cover page of each manuscript should contain the following data:

TITLE OF WORK
FULL NAME OF AUTHOR
ADDRESS
PHONE NUMBER

Each page of the manuscript and program listing should include the author's name, the title of the work, and the page number in the upper right hand corner.

The article and any accompanying program **should be submitted as a standard text file on a DOS 3.3 disk.** Label the disk with the title of the work and the author's full name and address. On disk, text must be single-spaced only. Please identify your editing program.

Original disks are always returned as soon as possible. Other materials will be returned only when adequate return packaging and postage is enclosed. We are not responsible for unreturned submissions. We *will guarantee* the return of original commercial disks mailed to us for verification of an accompanying softkey.

You will be notified of the status of your submission within 4 to 6 weeks after it is received if the article is a softkey accompanied by an original disk. Please submit completed manuscripts directly; do not query first. Previously published material and simultaneous submissions are not accepted.

SUBJECTS

We prefer material on these topics:

1) Original program/article combinations
2) General articles (Apple computing)
3) Softkeys
4) Advanced Playing Techniques (APT's)
5) Hardware modifications
6) DOS modifications
7) Product reviews (hardware and software)
8) Utilities
9) Bit Copy Parameters

WRITING YOUR ARTICLE

Observe the following points of style:

A. Always assume that your reader is a novice and explain all buzzwords and technical jargon. Pay special attention to grammar and punctuation; we require technical competence but also good, readable style.

B. Whenever appropriate, a list of hardware and software requirements should be included at the beginning of the manuscript. When published, this list will be offset from the main text.

C. Include the name and address of the manufacturer and the price when a commercial program is mentioned. This is of particular importance in PRODUCT REVIEWS.

D. When submitting programs, first introduce the purpose of the program and features of special interest. Include background information describing its use. Tips for advanced uses, program modifications, and utilities can also be included. Avoid long print statements and use TABs instead of spaces.

Remember: A beginner should be able to type the program with ease.

E. A PROGRAM is not accepted for publication without an accompanying article. These articles, as well as articles on **hardware** and **DOS modifications** MUST summarize the action of the main routines and include a fully remarked listing.

F. GENERAL ARTICLES may include advanced tips, tutorials, and explorations of a particular aspect of Apple computing.

G. SOFTKEYS of any length are acceptable and must contain detailed step-by-step procedures. For each softkey, first introduce the locking technique used and then give precise steps to unlock the copy-protected program. Number each step whenever possible. We accept articles which explain locking techniques used in several programs published by the same company.

H. When altering game programs, the changes made are sometimes extensive enough to warrant the title of ADVANCED PLAYING TECHNIQUE (APT). APTs can deal with alterations to a program, deleting annoying sounds, acquiring more points in play and avoiding hazards. Again, provide step-by-step instructions to complete each APT and explain each step's function. APT's of 100 words or more are preferred.

AUTHOR'S RIGHTS

Each article is published under the author's byline. As a rule, all rights, as well as one-time reprint rights are purchased. Purchase of exclusive rights to programs is required; however, alternate arrangements may be made with individual authors depending on the merit of the contribution.

PAYMENTS

COMPUTIST pays upon publication. Rate of payment depends on the amount of editing required and the length of the article. Payment ranges from $20 to $50 per typeset page for an article. We also pay $10 to $20 for short softkeys and APT's. A fully explained softkey accompanied by the commercial disk for verification may earn up to $50 per typeset page.

Please mail your submissions to:

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

FREE

your disks from their uncopyable status

Now you can make necessary backups of your locked-up commercial software with

The BOOK of Softkeys

shows you how to softkey (undo copy-protection on) commercial software.

Volumes II and III are being compiled now!

Volume I
(157 pages)
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.

I want ____ copies of The Book Of Softkeys Volume 1. I have enclosed $20 per book. Foreign orders add 20%. U.S. funds drawn on U.S. banks. Washington state orders add 7.8% sales tax. Send your orders to: Softkey Publishing; PO Box 110816-T; Tacoma, WA 98411

Name ________________________________

Address ________________________________

City ____________ State ______ Zip ______

Country ____________ Phone ____________

VISA ____-____-____-____ Exp. ______

Signature ________________________________