For The Serious User Of Apple][Computers

COMPUTIST

Issue No. 45 July 1987

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

Softkeys For:

Sands of Egypt
Agent U.S.A.
Kindercomp
Wavy Navy
Raid Over Moscow
Crime Stopper
Final Conflict

Feature:

Write Protecting the Microsoft RAM Card

(Page 20)

Core:

Copy Master: a review

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

BULK RATE
U.S. Postage
PAID
Birmingham, AL
Permit No. 740

Coping With COMPUTIST

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

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

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

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

PR#6

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

Control characters are specially boxed:

6[CTRL]P

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

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

■ **Software Recommendations:**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10HOME:REMCLEAR SCREEN

The LIST will look like:

10 HOME : REM CLEAR SCREEN

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

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

CALL -151

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

When finished, return to BASIC with:

E003G

BSAVE the progra
address and
article.

■ **Keying I**
is printed to
To key it in,

Without this assembler, you will have to translate pieces of the source code into something *your* assembler will understand. A table of S-C Assembler directives appears in COMPUTIST 17.

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

■ **CHECKSOFT Instructions:**

LOAD *filename*
BRUNCHECKSOFT

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

■ **CHECKBIN Instructions:**

CALL -151
BLOAD *program filename*

Install CHECKBIN at an out of the way place

BRUN CHECKBIN,A$6000

Get the checksums by typing the starting address, a period and ending address of the file followed by a [CTRL]Y [RETURN].

xxx.xxx[CTRL]Y

Correct the lines at which the checksums differ.

You have a LEGAL RIGHT to an unlocked backup copy

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

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

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

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

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

Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along
prepared, only as part
in the program.
only with the

(C 117)

Be assured of receiving the latest issue of COMPUTIST each month without the hassle of making a trek to the local computer store and not finding COMPUTIST so of course you ask the clerk if they have COMPUTIST and they tell you they don't carry COMPUTIST but maybe you could try the computer store down the block because they might have an issue of COMPUTIST so you go to the computer store down the block and ask them if they have COMPUTIST but of course they don't (they just ran out of COMPUTIST yesterday) so you ride your unicycle clear across town to see if the computer store across town has COMPUTIST but when you go inside and ask for COMPUTIST they don't have COMPUTIST either and suggest that you try the store you originally went to in the first place so you end up going home disgusted that you missed another issue of COMPUTIST.

Can't Find COMPUTIST Anywhere?

Stop searching and **subscribe now!**

annual subscription rates (12 issues):

U.S. Subscriptions - sent third class - **$32**
U.S./Canada/Mexico - sent First Class - **$45**
U.S./Canada/Mexico First Class PLUS library disk - **$100**

All other Foreign Subscriptions - **$75**
All other Foreign PLUS library disk - **$140**

■ Use the form on the right to order or renew your subscription.

■ You may upgrade your current subscription to a magazine + disk combination by sending $5.50 ($6.50 foreign) per remaining issue.

■ Check your mailing label to see if you need to renew your subscription.

■ If you're moving, let us know at least 30 days in advance.

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

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

Yes, I want to subscribe to COMPUTIST.
Enclosed are funds for an annual (12 issue) subscription.

I am ☐ **A new Subscriber**
☐ **Renewing my current subscription**
☐ **Changing my address (please include latest label)**

☐ **U.S. - $32**
☐ **U.S./Canada/Mexico First Class - $45**
☐ **U.S./Canada/Mexico First Class plus Library Disk - $100**

☐ **All other Foreign - $75**
☐ **All other Foreign plus Library Disk - $140**

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA MC _____-_____-_____-_____ Exp._____

Signature _____ CP45

U.S. Funds drawn on U.S. bank. Please allow 4 to 8 weeks for subscription to commence.

Send orders to: **COMPUTIST PO Box 110846-T Tacoma, WA 98411**
(206) 474-5750

Is it time to renew? Are you moving soon?
Take time now to save time later!

40¢ per disk?

And that INCLUDES the Tyvek sleeve, too?

DS/DD
5 1/4 "
for use with:

- **Apple**
- **Commodore**
- **Atari**
- **Epson**
- **TRS-80**

50 disks for only $20

- Sold only in sets of 50.
- 100% guaranteed.
- reinforced hubs.
- Tyvek sleeves & Write-protect tabs included.

Please send me _____ sets (50 disks per set) of 5¼" DS/DD diskettes. I have enclosed $20 PLUS $4 shipping & handling for each set and made checks / money orders payable to SoftKey Publishing.

- 50 DS/DD diskettes for $20.

Sets include Tyvek sleeves, write protect tabs and labels.
- Enclose $4 shipping & handling for each set.
- U.S. and Canadian orders only. Please include street address for UPS delivery.
- Washington state orders add 7.8% sales tax.
- US funds drawn on US banks.

Offer good while supplies last

Name _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA MasterCard _____-_____-_____-_____ Exp. _____

Signature _____ CP45

Send your order to:

Disk Offer SoftKey Publishing PO Box 110846-T Tacoma, WA 98411

(206) 474-5750

Come And Get It!!

$17.95

The Book of Softkeys Volume III

Featured in **The Book of Softkeys Volume III** are:

Alien Addition • Alien Munchies • Alligator Mix • Computer Preparation SAT • Cut And Paste • Demolition Division • DLM (Developement Learning Materials) software • EA (Electronic Arts) software • Einstein Compiler version 5.3 • Escape From Rungistan • Financial Cookbook • Flip Out • Hi-Res Computer Golf II • Knoware • Laf Pak • Last Gladiator • Learning With Leeper • Lion's Share • Master Type v1.7 • MatheMagic • Minus Mission • Millionaire • Music Construction Set • One On One • PFS software • PS (Penguin) Software • The Quest • Rocky's Boots • Sabotage • Seadragon • Sensible Speller IV • Snooper Troops II • SoftPorn Adventure • Stickybear series • Suicide • TellStar • Tic Tac Show • Time Is Money • Transylvania • Type Attack • Ultima III Exodus • Zoom Graphics • Breaking Locksmith 5.0 Fast Copy • Csaver • The Core Disk Searcher • Modified ROMs • The Armonitor

We've been working hard to compile

The Book of Softkeys Volume III.

HURRY!

Order yours now!

Please send _____ copies of The Book of Softkeys - Volume III. Enclosed is $17.95 plus $2 ($5 Foreign) for shipping and handling per book. Washington state residents add 7.8% sales tax. Send check/money order to:

SoftKey Publishing
PO Box 110846-T
Tacoma, WA 98411

Name _____ ID# _____

Address _____

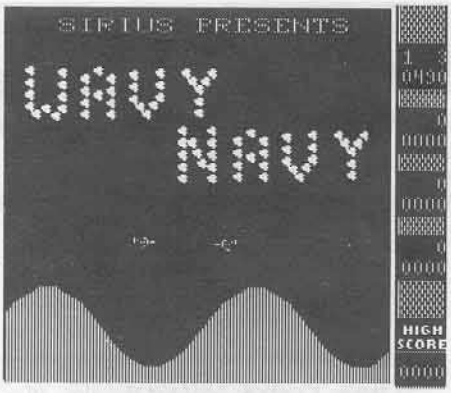
City _____ State _____ Zip _____

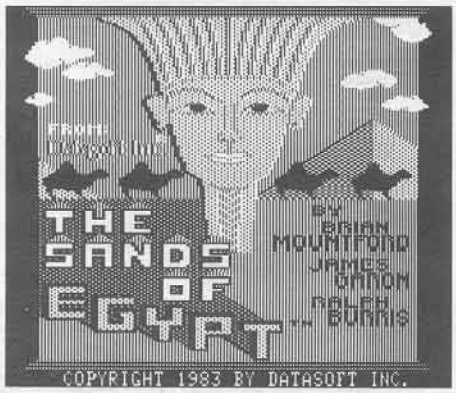
Country _____ Phone _____

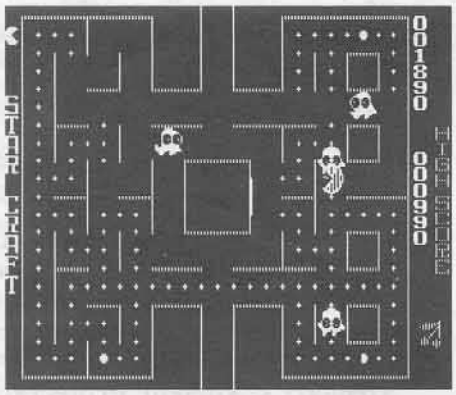
VISA MasterCard _____-_____-_____-_____ Exp. _____

Signature _____ 45

US funds drawn on US bank







This month's cover:
Graphics from Datasoft's "Sands of Egypt."

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

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

Entire contents copyright 1986 by SoftKey Publishing. All rights reserved. Copying done for other than personal or internal reference (without express written permission from the publisher) is prohibited.

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

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

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

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

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

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

COMPUTIST

Issue 45 July 1987

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

softkeys:

12 Mouse Calc
by Richard Glasgow

16 Sands of Egypt
by Steve Ellis

17 Number Farm
by Scott McCandless

18 Agent U.S.A.
by Bill Johnson

20 Wavy Navy
by Rich Etarip

23 Kindercomp
by B. Croome

24 Flight Simulator Update
by Eric Sunshine

features:

10 Write Protecting the Microsoft RAM Card
This article shows you how to turn a mild mannered RAM card into a useful cracking tool.
by Philip Goetz

12 Keys to Success on the Franklin Ace
With a new PROM, your Franklin Ace 100 keyboard becomes fully enhanced.
by Kevin D. Miller

14 Modified F8 ROMs On the Apple ///
Yes, even Apple /// users can Reset into the monitor. That is, after they follow the procedure outlined in this article. *by Eric Deslauriers*

core:

11 An Owner's Review of The Copy Master II
A look at the new cracking card from Applied Technology Laboratories. *by Andrew Swartz*

departments:

4 Input
6 Bugs
7 Readers' Softkey & Copy Exchange
25 The Product Monitor

input

Please address letters to:

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

Include your name, address and phone number.

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

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

Opinions expressed are not necessarily those of COMPUTIST or SoftKey Publishing.

Pandora's Box Update

First of all, I wish to thank you for a super magazine. I also would like to thank all those persons who let us share their knowledge by sending in their softkeys and giving us hints on how to deprotect or copy programs ourselves.

How did we ever do without it? I remember all too well, like that time back in 1983, I had bought my brand new Apple][Plus only a few weeks before and was trying to initialize a data disk with my Apple Writer II program and promptly initialized the masterdisk. Then after waiting for more than 6 weeks for a replacement, (Apple was moving from Sunnyvale to Cupertino) I had to go and pick it up at the local customs office, where the officer tried hard to charge me duty on the value of the program which Apple very generously sent me without charging a penny. Well, I did not have to pay duty, after some arguing and pointing out that I already payed duty and sales tax etc, on the original program.

I had learned my lesson, "Write-protect the master and always make a backup of every program you get AND only use the backup."

I have received all the back issues, Softkey books volume I and II, and disks that I ordered. Thank you for looking after me so promptly.

Here is an update on deprotecting Pandora's Box to a COPYA copy.

Pandora's Box

Datamost, Inc.
9748 Cozycroft Ave.,
Chatsworth, CA 91311

Requirements:
COPYA
Blank disk
Sector editor
Original Pandora's Box

After reading Clay Harrell's deprotection of Pandora's Box (COMPUTIST No. 6, page 5) I knew that my version was different, the VTOC and catalog is not on track $11 and my version also did a nibble count while booting up.

After several tries to make a backup with my nibble copiers without success I decided to try to deprotect this game.

After some snooping around with my Copy][Plus (v5.5) sector editor, in the patched DOS 3.3 mode, I was able to find a VTOC and catalog on track $5. The only file in the catalog was named ..... and the VTOC looked all messed up. On track $0, sector $3, I found that DOS was told to look for a DF in the epilogue and the rest of DOS looked almost normal.

So my next step was to make a COPYA copy and use my trusty Copy][Plus sector editor to change some of those JMP's (one at a time). Get into the monitor and do some snooping around in memory. While in the monitor, I found some of the tricks they used trying to stop us from deprotecting the game, such as loading the addresses of last BLOAD length, last BLOAD start and catalog track number with the value FF. On the disk it showed up like this:

```
LDA #$FF
STA $AA60
STA $AA61
STA $AA72
STA $AA73
STA $AC01
```

Also after reconnecting DOS the program would fill the screen with @'s and would hang up, the only way out was to power down.

After several tries, I got the game to load until just past the nibble count and it jumped into the monitor with the final address $235, I scanned the disk for a JMP or JSR to an address before $235. I was lucky, on my first try I found a 4C 30 02 on track $22, sector $D. I changed this to an EA EA EA, and it worked.

In a Nutshell

1) RUN COPYA

2) When COPYA asks for the slot, stop the program and type the following:

```
[C]
70
CALL-151
B942:18
3D0G
RUN
```

3) Follow the prompts to make a copy.

4) After the disk is copied, use a sector editor to make the following changes on your copy.

TRACK	SECTOR	BYTE	FROM	TO
0	3	42	38	18
0	3	91	DF	DE
22	D	18	4C	EA
22	D	19	30	EA
22	D	AA	02	EA

Don't forget to write the sectors back after changing them!

Leo and Eric Van Der Loo
Ontario, Canada

Airheart's Ancient Protection

For those of you who have not noticed, the hottest new game out for the Apple is Airheart, by Broderbund software. The game features superb double-hi res graphics and great animation. The folks at Broderbund, however, are following a growing trend to ressurect ancient copy protections. Upon inspection of my Airheart disk, I discovered that the protection greatly resembled the one on Choplifter, another game by Broderbund. Here is a simple way to obtain a working backup with Copy][Plus: Copy tracks 0-22 with manual bit copy. Next, copy the disk again over what you just copied using the Choplifter parameter. Good Luck! Also, if anyone knows how to copy/deprotect EDD IV, please write me a letter telling me how.

Eric Redlinger
Reeds Bridge Rd.
Conway, MA 01341

input

Various Notes

When is COMPUTIST going to go to a weekly or even bi-monthly format? I just can't stand waiting a whole month for new information. Actually, I am writing to share a little information!

In COMPUTIST No. 39, the softkey for Cross Country Race worked fine. For those of you who have Countdown to Shutdown, also by Activision, try this:

TRACK	SECTOR	BYTES	TO
0	A	DB-FF	EA's
0	B	00-1A	EA's
0	B	1B	A9
0	B	1C	FF

In COMPUTIST No. 33, the softkey for Hardball failed on my copy, The author pointed out some funny business going on at B260, so I tried EA EA EA starting with that byte. Bring on the hot dogs and beer!!!

The Skyfox update was no help, but Copy][Plus v 5.5 Auto-Copy will render a fully working copy complete with Space Invaders and full Mockingboard sound!

Going back to COMPUTIST No. 32, my version of Cubit starts at 4063 and the number of men is at 4014.

Regarding the recent letters on the piracy controversy, I wonder who the real pirates are. I find it a little upsetting when a program I paid $25 for two months ago is now selling for a mere $10. I also wonder why an identical program for another computer always sells for less. I don't know about anyone else, but I would certainly own more original software if the prices were a little more realistic. You'd think that one Ferrari per executive would be enough!!!

In closing, I would like, as Jim Hart did, to correspond with others of like interest, we already read the same great magazine.

M. Ferreira
5380 Marigold Lane
Santa Rosa, CA 95401

Turbocharged Logo

Your magazine provides a real service to those of us trying to get the best value from our machines. Your softkey for Terrapin Logo v2.00 in COMPUTIST No. 35 turns a laboratory curiosity into a usable, and useful, programming language. However, it can be made even better by the addition of a "speed DOS" without any "judicious surgery to the SDOS file". Here is my experience with fast DOS:

Time to boot up from a "BRUN SDOS" command

DOS 3.3	40 seconds
DOS 3.3 + Speedload	35 seconds !!!???
DaviDOS	35 seconds
DiversiDOS	Doesn't load
ProntoDOS	9.3 seconds

DaviDOS only took 35 seconds, but it crashes when trying to do a CATALOG (or anything else for that matter).

So, to turbocharge Terrapin Logo, simply boot it up under ProntoDOS, or initialize a broken-out disk with ProntoDOS. The big puzzle is why DOS 3.3 plus Speedload had no apparent effect.

Kevin W. Clark
Beaverton, OR

TASC Comment

I am writing in response to a letter you published in the Input section of COMPUTIST No. 31 called "A difficult TASC."

I can well understand why Microsoft would be reluctant to give out source code for TASC. According to my sources, TASC was originally written in Applesoft BASIC and then used on itself! The only program that would have an assembly source code listing is the library module called RUNTIME. I ask you- would you want your basic source code to such a powerful program floating around in public hands? It would be nice for some of us, to be sure, but you know that somebody out there would try to modify it a little bit and then sell it as their own. I know that I, as an author and avid COMPUTIST reader, would not want my source code to a program published for public use and, lets face it, plagiarism. I fully sumpathize with Mr. Brand's problem - I have had the problem a few times myself, but under the circumstances I can also see that Microsofts stand on the matter is correct. Best of luck on creating an UNTASC program. The person who successfully does so will make his fortune.

Mike Crane
Provo, UT

Super Boulder Dash

While reading my latest isue of COMPUTIST. I noticed Super Boulder Dash listed on your "Most Wanted List". I have found that Super Boulder Dash can be easily deprotected using Randy Abel's Boulder Dash I and II softkey in COMPUTIST No. 38. I'm sure your readers would appreciate it if you would clarify this error.

Jerrell Watts
Odessa, TX

Flight Simulator Troubles

First of all, I hope you can answer my letter or at least give this to Eric Sunshine, author of the Flight Simulator Softkey (A2-FS2), because I have a doubt in the deprotection of A2-FS2. I have the A2-FS2 versions 1.05 and 2.0 and I can't capture the language card data in both versions. Steps 1 through 6 work fine, steps 8 through 16 also work well but... when I execute step 7, command ACAEG that stands in code like this:

ACAE:20 92 1D JSR $1D92

That is the code for BOOT2 step 5. I find that the code between $1D37 through $1E00 are empty, resulting that this instruction among others in the code of the main part of boot at $A7E0-$B2DF can't execute via JSR to $1D92 because this JSR doesn't exist only zeroes.

I finished the method step-by-step through the end bypassing step 7 and all is in place but the language card data and I don't know how to capture it.

When I boot by copy of A2-FS2 it boots okay but after loading the title LOADING.... and try to load language card data in tracks 20, 21, and 22 stops without stopping the drive motor and hangs up.

Also, I have found that the softkey for Fantavision (COMPUTIST No. 30) did not work for my version of this program because I had different address epilogues. To fix this, I changed line 1130 of the controller to:

1130 DATA 255,255,255,255

and that's all, the softkey works perfectly.

Manuel Sanabria Pastrana
Mexico

Mr. Pastrana: *Eric Sunshine's reply to your letter can be found on page 28 of this magazine.*

input

Hacker II, Robotwar, Castle Wolfenstein, Silent Service

In "Hacker II", the protection is based on a nibble count at track $23. Track $23 is very difficult to copy (drive speed critical). Can't make a backup ?!?! Only way to back up this program was to crack it. (At least for me with my poor disk drive) To bypass the nibble count check we have to modify file HACKER II HELLO. Make the following changes to HACKER II HELLO:

BLOAD HACKER II HELLO
CALL -151
63C9:A9 55 8D 35 64 60
3D0G
BSAVE HACKER II
HELLO,A$6000,L$0589

For those of you who are interested in what's going on:

```
63C9: LDA #$55  LOAD NIBBLE COUNT RESULT
63CB: STA $6435 PUT IN THE RIGHT LOCATION
63CE: RTS       AND RETURN
```

Let's hope that there are no more nibble count checks.

If you carefully examine the disk you will find a text file that has some hints. You can use FIXCAT (from Bag of Tricks) to do this. Just scan for the lost files and when you see some normal text, recover the file.

I have seen softkeys for Robotwar and Castle Wolfenstein, but here is some information about DOS used on those programs (there are at least two different versions of Castle Wolfenstein) so this scheme is for the oldest one.

Both of these are using 13-sector based systems. To easily copy the files to 16-sector diskettes, we can use Muffin with the following modifications:

BLOAD MUFFIN
CALL -151
1AA9: 18 66 2D 60
801G

After these transfers, programs will not work as expected. We have to make some changes to the DOS. For Robotwar we must make a short patch to the DOS. Just make your own startup program, as follows:

NEW
10 POKE 40288, PEEK(1): POKE 40289, PEEK(2)
20 PRINT CHR$(4);"RUN HELLO"
SAVE HELLO ROBOT WAR

And use Master Create (or Copy][Plus) to set this as startup program.

This must be done because all the files are written as "Applesoft files." But really they are machine language files. If we don't make the patch to DOS the program will simply not run. Because when you load an Applesoft file DOS jumps to Applesoft BASIC that tries to make corrections to the current program. Since in this case the program is in machine language the BASIC will destroy parts of the machine language program.

Castle Wolfenstein does not need the previous patch.

The next step is to make the following changes to the programs. Use a sector editor (that has search capability) to make some "corrections". You shouldn't find these from DOS.

CHANGE	TO	COMMENT
7E9E	BD9E	Different terminal output routine addresses.
A90C8DEDB7	A90E8DEDB7	Catalog sector begins from different sector.

That should do it! Now you can insert some of those fast DOSes on your disk and enjoy!

I'm having a hard time with my Apple Writer 2.0. Does anyone have any tips?

Also I noticed a bug in the Silent Service softkey: Should byte 9A be A9?

Jussi Eloranta
Lexington, MA

The Bard's Attributes

Joe Montano's, "The Bard's Dressing Room", COMPUTIST No. 36, asked for an explanation of how the attributes are stored in bytes $10-$13 of the character's record. Here is how it is done.

Let's assume your super hero character has the following attributes, St = 8, Iq = 14, Dx = 15, Cn = 9, and Lk = 13, (some super hero). You use your favorite sector editor and find in bytes $10-$13 the following values, $43, $8F, $4B, and $4E. At first this all looks very confusing, but soon it will all make sense. The method used is similar to the packed variable storage of Pascal.

First lets break the hex bytes into their bit components.

```
  $4 3      $8 F      $4 B      $4 E
0100 0011 1000 1111 0100 1011 0100 1110
```

Now realizing the apparent maximum attribute value is 18, or 0001 0010, we see only five bits are needed and the leftmost three bits will never be used. For every two bytes, sixteen bits, we can store three five bit attributes and have one bit left over. Inspecting several characters quickly showed which bit was not being used. Distributing the above bits into groups of five and including a skip bit in the appropriate place, we get the following sequence.

```
  St      Iq      Dx      Cn      Lk      ??
01000  01110 0 01111  01001  01101 0 01110
  8       14      15      9       13      14
```

I do not know what the last attribute is. I know it is not a checksum and it may not be even used in the game. Now if you want all your super hero's attributes to be 18's, or 10010's, here's how to go about it.

The following bit pattern is needed for all 18's

```
10010  10010 0 10010  10010  10010 0 10010
```

Combine these into four bytes.

```
1001 0100  1001 0010  1001 0100  1001 0010
  $9 4       $9 2       $9 4       $9 2
```

Using your favorite sector editor, change bytes $10-$13 to the hex values 94, 92, 94, and 92. Boot up Bard's Tale, add your character with his usual team and resave to disk. That's all there is to it. By the way, if your style is a really lopsided game, you can make your super hero's attributes all 31's, or 11111's. Eighteen is not the maximum value an attribute can have.

Thomas Kazmierczak
Clifton, NJ

bugs

COMPUTIST No. 37:

DOS Alterer:

Line 1850 of the program should read:

```
1850 HTAB 6 : VTAB 7 : PRINT "FILENAME^ =>"
     ;N$
```

Book Of Softkeys I:

Bag Of Tricks:

The BLOAD in line 210 of the Hello Program should be a BRUN.

readers' softkey & copy exchange

Winston Ou's softkey for...

Raid over Moscow

Access Software, Inc.

Requirements:
Any simple copy program
A sector editor

Raid Over Moscow is a typical anti-Soviet arcade game based on this scenario: three years after the signing of the SALT IV treaty which provided for total disarmament of nuclear weapons, a Persian gulf incident causes the Russians to launch a nuclear strike. Yes, the evil Russians have secretly kept a cache of nuclear missiles. Tsk, tsk tsk. It is up to you, commander of a space station of Stealth Bombers, to raid the major Soviet cities and destroy the missile control systems, thus preventing the missiles from exploding (huh? Well, that's how the story goes). After that, go destroy the Soviet defense center in Moscow as a little slap on the wrist for the evil creatures.

The first time I tried to back up this program I used Disk Muncher 1.0, which seemed to work fine and dandy until, a few seconds after booting up, I heard an awful swishing noise with a "click" on the end of it repeated quite a few times. This was the tragic sound of a disk drive hopelessly searching for something on track 0. In any case, none of the bit copiers I tried worked, so I decided to be intelligent and unprotect it.

The entire disk was written with something very similar to DOS 3.3, so it was easy enough to examine track 0, sector 0, which eventually jumped to $BB3E, which jumped to $BB00, where the fun began. Here's the code:

```
BB00-  A9 00      LDA #$00
BB02-  8D 78 04   STA $0478
BB05-  A6 2B      LDX $2B
BB07-  A9 00      LDA #$00
BB09-  20 A0 B9   JSR $B9A0
BB0C-  20 44 B9   JSR $B944
BB0F-  A5 2D      LDA $2D
BB11-  C9 00      CMP #$00
BB13-  D0 F7      BNE $BB0C
BB15-  A9 01      LDA #$01
BB17-  8D FF B7   STA $B7FF
BB1A-  AD FF B7   LDA $B7FF
BB1D-  0A         ASL
BB1E-  20 A0 B9   JSR $B9A0
BB21-  20 44 B9   JSR $B944
BB24-  A5 2D      LDA $2D
BB26-  AC FF B7   LDY $B7FF
BB29-  D9 30 BC   CMP $BC30,Y
BB2C-  D0 D7      BNE $BB05
BB2E-  C8         INY
BB2F-  98         TYA
BB30-  C9 23      CMP #$23
BB32-  90 E3      BCC $BB17
BB34-  A9 00      LDA #$00
BB36-  20 A0 B9   JSR $B9A0
BB39-  4C B3 08   JMP $08B3
```

As I can now see, most of this code is innocent enough: the JSRs to $B9A0 and $B944 are the move disk arm and read address field routines, respectively. However, when I placed a $4C 59 FF (JMP into monitor) instruction at $BB39 on my work copy, nothing happened, so I assumed that the program never reached that point and that the check had to be somewhere before there. Then I noticed that at $BB2C it had a branch that went all the way back to the beginning of the routine if two addresses didn't match up. So, I changed the second part of the BNE (branch if not equal) instruction to $00, causing the program to continue even if the addresses were unequal.

And that was it! One lousy little byte... anyway, I played the game through once and watched the three line message of approval at the grand finale; it worked fine.

Now that I've bored you all to death, here are the steps:

1) Copy the disk with your simple copy program.

2) Get out your sector editor, and edit Track $0, Sector $5, byte $2D. Change it from $D7 to $00.

Marc Morrisette's softkey for...

Crime Stopper

Hayden Software

Requirements:
Super IOB 1.5
A blank disk
Crime Stopper disk

Crime Stopper is an old text adventure game by Hayden, unfortunately copy-protected. However, the protection is quite minimal and easy to crack.

Whenever I buy a new program, I first boot it up and play it and then attempt to do my routine backup. The disk, when booted, displays an Applesoft prompt (]) on the screen, so it would seem that the disk uses a modified version of DOS. After trying my luck with Copy][Plus 5.0, I booted up the backup. No luck. The disk would boot most of DOS, but just before displaying the Applesoft prompt, the computer would display a character on the lower right corner of the screen and return me to BASIC.

If I booted my original disk, and after booting completed, inserted my backup, the program portions would load properly. I decided that this was not good enough, since I wanted a copy that would boot. I then tried using COPYA, after turning off DOS's error checking routine (B942:18 in the monitor). COPYA kept giving me the "UNABLE TO READ" message. I was overjoyed when I saw the softkey for Crystal Caverns in COMPUTIST No. 31. (I also own Crystal Caverns and Micro Addition, both with the same protection scheme as Crime Stopper.) I tried the softkey on all three of these.

No luck! I had another version of MUFFIN, and the protection on my Crystal Caverns was different! I decided to snoop a little further. After using the Nibbler (from COMPUTIST No. 19) to view the raw track nibbles from the disk, I was able to confirm the following. Tracks $0 through $2 (the DOS tracks) have the address and data epilogues changed from the normal DE AA to ED AA. Tracks $3 through $4 and $6 through $22 (the tracks which contain the program files) have changed the address prologue from the normal D5 AA 96 to BB AA 96. Track $5 is blank (full of $FFs).

I then wrote up a controller for Super IOB 1.5 to read tracks $3 through $4 and $6 through $22 using an address prologue of BB AA 96 and write using the normal D5 AA 96. I INITialized a blank disk using "HELLO" as the greeting program name. Next, I installed the controller (printed at the end of this article) into Super IOB 1.5 and ran it. Amazingly, it worked! Fortunately, the program does no check for the original disk anywhere in it. Now, Crime Stopper is open to the rapidly growing hobby of APT.

Step by Step

1) INITialize a blank (or unwanted) disk using "HELLO" as the greeting program name.

```
INIT HELLO
```

2) Install the controller printed at the end of this article into Super IOB 1.5 and run it. Answer "NO" to the format option.

3) Copy the original Crime Stopper disk to the INITialized disk.

4) Enjoy the deprotected Crime Stopper!

This softkey will work on any Hayden program in which the following is true: Tracks $0 through $2 have the address and data epilogues changed from DE AA to ED AA. Tracks $3 through $4 and $6 through $22 have the address prologues changed from D5 AA 96 to BB AA 96. Track $5 is blank (full of $FFs). I have tested this softkey (successfully) on the following Hayden programs: Crystal Caverns, Crime Stopper, and Micro Addition. I know there are others that it will work on.

controller

```
1000 REM CRIME STOPPER CONTROLLER
1010 ST = 15 :LS = 15 :CD = WR :FAST = 1
1020 TK = 3 :LT = 5 : GOSUB 1050
1030 TK = 6 :LT = 35 : GOSUB 1050
1040 HOME : PRINT "COPYDONE" : END
1050 RESTORE : GOSUB 190 : GOSUB 490 : GOSUB
     610
1060 GOSUB 230 : GOSUB 490 : GOSUB 610 : IF
     PEEK (TRK ) = LT THEN RETURN
1070 TK = PEEK (TRK ) :ST = PEEK (SCT ) : GOTO
     1050
5000 DATA 187 ,170 ,150
```

controller checksums

```
1000  - $356B     1050  - $A7D7
1010  - $88AA     1060  - $F197
1020  - $9392     1070  - $3F2F
1030  - $FB01     5000  - $7432
1040  - $26F8
```

Jim S. Hart's softkey for...

Key Perfect 5.0

MicroSparc, Inc.
45 Winthrop St.
Concord, MA 01742
$29.95

Requirements:
Original Key Perfect 5.0 disk
One blank disk
A copy program that ignores unreadable sectors (Locksmith Fast Copy, etc.)
ProDOS User's Disk or equivalent
A good sense of humor

Key Perfect 5.0 is a program designed to go through your Applesoft or machine language programs and generate a checksum table (sound familiar?). This may be used to check against what you have typed in from *Nibble* magazine to see if there are any typos. My two gripes against the program: (1) it is protected, and (2) a program such as this should be provided free of charge, or for a nominal fee ($10 or so). *Nibble* has quit printing checksum tables with their older program, Apple Checker 3.0, so this FORCES users to buy the Key Perfect program to guard against typos. I'm glad COMPUTIST hasn't taken this policy. I always have several 'bugs' in the programs I type in so a checksum program is an essential.

The Key Perfect disk itself is ProDOS based so the protection usually isn't that hard to figure out (Spellworks and Word Juggler were exceptions). The reason you need a good sense of humor is that the protection scheme in Key Perfect is so simple it's funny! There is no direct access code on the disk that looks for anything other than normal 16 sector address marks so I decided to take a look at the main system file KEYPERF.SYSTEM. Here is the code at the start of this file:

```
2000-  LDA $BF30    ;Get slot no.
2003-  STA $2021    ;Store it in parm list
2006-  JSR $BF00    ;Call MLI
2009-  $80          ;Read block
       $20,20       ;addr of MLI parm list
200C-  BNE $2013    ;Was block readable?
200E-  RTS          ;Yes, bomb into monitor

2013-  JMP $6500    ;Else start program

2020-  $03,60,00,   ;MLI parm list
       $30,17,01
```

For a complete explanation of what all of this does I suggest that you read Worth and Lechner's *Beneath Apple ProDOS* (chapter 6). In a capsule, block $117 is read, and if an error occured (like an I/O error, perhaps?) execution is transferred to the main code and all is well. It so happens that block $117 is the last block on the disk, which means it resides on track $22. When you copy the disk with Locksmith Fast Copy, notice that track $22 produces a column of inverse "A's", in this case meaning that the track is unformatted. Therefore, if the read is successful then the disk is a copy and the code will go no further. This is easy to fix. Just replace the three bytes at $2000 from $AD 30 BF to $4C 13 20 (JMP $2013) and the check is jumped over!

Step by Step

1) Initialize a blank disk.

2) Copy the original onto the blank disk, ignoring the errors on track $22.

3) When the copying is done, take both disks out and put away your original.

4) Boot up ProDOS and get into BASIC.

5) Insert the copy disk and set the prefix to that of the copy.

```
PREFIX/KEYPERFECT
```

6) BLOAD the offending file, jump over the disk check, and BSAVE the file back to the disk.

```
BLOAD KEYPERF.SYSTEM, A$2000, TSYS
CALL -151
2000:4C 13 20
BSAVE KEYPERF.SYSTEM, A$2000, TSYS
```

8) You now posess an unprotected copy of Key Perfect to do your checksumming with. Wouldn't it be nice if all protections were this simple?

Jim S. Hart's softkey for...

The Final Conflict

Hayden Software
600 Suffolk St.
Lowell, MA 01854

Requirements:
The Final Conflict original disk
Demuffin Plus
A blank initialized disk
64K Apple II Plus or a way into the monitor

The Final Conflict is a wargame style program with the object being to (naturally) defeat either the computer or a human opponent. It is a relatively old game and as such I picked it up at a bookstore for $9.95. Using Locksmith 6.0's quickscan function along with the nibble editor revealed that the disk was DOS 3.2 formatted with an altered address prolog. Deciding to try a different approach than usual (i.e. Super IOB) I came up with the method below. Follow it to obtain a deprotected Final Conflict disk.

1) If you have a way to get into the monitor (Integer card, NMI card, etc.) go to step 5.

2) Boot a normal DOS 3.3 disk and get into the monitor:

```
CALL -151
```

3) Write enable the language card, copy the motherboard ROMs onto it, change the Reset entry point, and then write-protect the card:

```
C081 N C081
D000<D000.FFFFM
FFFC:59 FF
C083 N C083
```

4) Get back into BASIC.

E000G

5) Boot up the Final Conflict disk:

PR#6

6) When you see the Applesoft prompt, open the disk drive's door and go into the monitor by pressing Reset.

7) Move The Final Conflict's RWTS to a safe place:

6800<B800.BFFFM

8) Insert the blank initialized disk and boot it up:

C600G

9) Go into the monitor once again and BLOAD Demuffin Plus into place:

CALL -151
BLOAD DEMUFFIN PLUS, A$803

10) Disconnect DOS, move the Final Conflict RWTS back, and then reconnect the hybrid DOS:

FF59G
B800<6800.6FFFM
A851G

11) Start up Demuffin Plus:

803G

12) Copy all of the files from the Final Conflict disk to the blank initialized disk.

13) At this point, you have a deprotected disk. To speed things up you may want to add a fast DOS such as Diversi-DOS or Pronto-DOS. Enjoy your backup!

Richard Glasgow's softkey for...

Miss Mouse

Requirements:
Copy][Plus 6.5

Miss Mouse is a nice pre-school education program. The protection is more at the high school level. The program runs the file MIND1 which checks track three and if the correct information is found, the boot continues. Otherwise, a message stating that the disk is defective is displayed.

Step by Step

1) Use the sector copy of Copy][Plus and copy tracks $00-$02 and $04-$22.

2) Use the sector editor to change track $12, sector $0C, byte $1b from a $38 to a $60.

Peter Chan Shun Kit's softkey for...

Snoggle

Broderbund Software

Requirements:
48K Apple II
DOS 3.3 (or a fast DOS)
A blank disk
Snoggle diskette

Snoggle is a Pac-Man like game published by Broderbund Software in 1982. Since Snoggle is not a very good game and using a whole side of a disk to back it up is a wasteness, I decided to break it into a single file. Moreover, the protection scheme only relies on DOS 3.2 with altered DOS marks.

Now it is time to start boot code tracing the game.

1) Initialize a blank disk with no HELLO program.

INIT HELLO
DELETE HELLO

2) Enter the monitor by typing

CALL-151

3) Next, we want to move the Disk Controller ROM boot program to $9600.

9600<C600.CFFFM

4) Change the boot ROM to stop after loading the first sector and then run it.

96FA:98
9801:4C 59 FF
9600G
C0E8

5) The next stage jumps to $301. We will change the jump to $9301 and stop it there.

9800<800.8FFM
9805:98
9843:93
9301:4C 59 FF
9600G
C0E8

6) Now we move the next step up to $9300 and modify it to jump to a small routine at $9000. The routine checks the progress of Boot0 and enters the monitor when the RWTS has been read in.

9300<300.3FFM
9343:4C 00 90
9000:A5 3E C9 5D D0 03 4C 5D
9008:02 A5 3E 8D FE 90 A5 3F
9010:8D FF 90 4C 59 FF
9600G
C0E8

7) When we look at $90FE & $90FF, we can find that the next step will jump to $B700. Here we will modify the program to jump to the monitor after the game has been read in, not $5000 where the game starts.

CTRL-F
:00 60
B75F:4C 59 FF
B700G

8) Now Snoggle has been completely read in. We then move the code to a safe place and boot the disk with no hello program.

900<6400.AAFFM

(Insert the slave disk)

C600G

9) Enter the monitor and type in the following code.

CALL-151
5100:20 2F FB EE F4 03 A0 47
5108:A2 00 BD 00 09 9D 00 64
5110:E8 D0 F7 EE 0C 51 EE 0F
5118:51 88 D0 EC 4C 00 50

Check your typing against the following dis-assembly:

```
5100- 20 2F FB    JSR $FB2F
5103- EE F4 03    INC $03F4
5106- A0 47       LDY #$47
5108- A2 00       LDX #$00
510A- BD 00 09    LDA $0900,X
510D- 9D 00 64    STA $6400,X
5110- E8          INX
5111- D0 F7       BNE $510A
5113- EE 0C 51    INC $510C
5116- EE 0F 51    INC $510F
5119- 88          DEY
511A- D0 EC       BNE $5108
511C- 4C 00 50    JMP $5000
```

10) We also have to put a jump to this routine right before the starting address of the program.

8FD:4C 00 51

11) Now we can save the file on to the disk.

BSAVE SNOGGLE,A$8FD,L$5B03

Now you have a 93-sector BRUNable file of SNOGGLE. Have fun !

A Hardware modification:

Write Protecting the Microsoft RAM Card

by Philip Goetz

Requirements:
Microsoft 16K RAMcard
Light gauge insulated wire
Single pole/double throw (SPDT) switch
Soldering iron

Note: This procedure requires modification of your 16K RAM card, which will void its (and possibly the computer's) warranty. Neither the author or COMPUTIST will be responsible for any damages incurred as a result of this procedure. Observe all the usual (and some of the unusual) precautions when working with electronic equipment.

An important cracking tool is an old style F8 ROM which resets into the monitor, like those in the early Apple IIs. They let you stop a program at any time, so you can with luck BSAVE it, or at least capture its DOS.

You can copy the monitor into a RAMcard and change it to reset to the monitor, but most well-protected programs erase the RAMcard on bootup. Fortunately, a simple modification lets you write-protect your RAMcard with the flip of a switch. It requires a small SPDT switch (one which switches a contact in the middle between 2 other contacts) and 3 short pieces of wire. When the switch is in one position, the RAMcard will function normally. When it is in the other position, it becomes write-protected (ROM-like). To use it as a cracking tool, copy a ROM which resets into the monitor into it and flip the switch.

Programs which use 64K to run will not run when the card is write-protected, but you can reset to the monitor. Programs which don't need 64K may circumvent a write-protected RAMcard by switching to the ROMs, but this is unusual.

This will void your RAMcard's warranty, and possibly your computer's warranty. Neither I nor COMPUTIST are responsible for damages caused to your RAMcard, your computer, yourself, or your close relatives. This will only work on Microsoft or similar cards.

First fasten the switch to the RAMcard. I used plastic cement to glue it to the upper rear of the component side of the card. Position it as low as possible so the switch can be flipped through the slot in back of the Apple case without removing the lid. You may have to file the top of the slot on the lid (for a II+) to make the slot tall enough; on a IIe, you may have to use an external switch or remove the lid to switch it.

Drill a hole or file a depression large enough to run 2 wires through at the top of the card near the switch. The wires which run from one side of the board to the other will go through here so they won't be squeezed against the bottom of the lid. Make sure that 2 wires fit in it comfortably.

Locate resistor R3, which is in the center of the RAMcard board. On my clone of the board it is called RJ. Solder a wire on the back of the card to the top of R3. Run it through the hole to the front, and solder it to the contact on the switch which will be open when the switch is in normal position.

Find chip U18 (a 74LS00). It is at the bottom near the rear of the card. Gently remove it using a chip-puller or small screwdriver. There will be a notch or mark at one end of the chip. The pin to the left of the notch is pin 1. Count two down to find pin 3. Bend it out perpendicular to the other pins so it will not touch the socket when reseated. Reseat it in its socket, making sure that the notch points the same way as the notches on all the other chips. Solder a wire to pin 3, which is sticking out, and solder the other end to the contact on your switch which will normally be closed.

Locate chip U16 (the only 74LS175). It is two chips above U18. Solder a wire to the contact for pin 13 of chip U16 on the back of the board. Run it through the hole at the top of the card and solder it to the switch's middle contact. Pin 13 is found by counting from pin 1 down to pin 8 on the end of the chip, switching sides, and counting back up towards the notch to 13. It is the fourth pin down on the right side of the notch. The switch should now be wired like this:

```
                          ----------
Top of resistor R3        o--|        |
                             | Open   |
Pin 13 of U16 (74LS175)   o--|        |
                             | Closed |
Pin 3 of U18 (74LS00)     o--|        |
                          ----------
```

Now the card will operate normally when the line from chip U16 to U18 is closed, and will be write-protected when the line from R3 to U16 is closed. The red write-enable LED still reflects the card's status. Setting the switch to write-protect guarantees that the card cannot be write enabled, but if the card is write enabled when you flip the switch it will remain so until you send another command to the RAMcard. Thus disable writing with, say, a $C080 before booting your protected disk.

The following text file will copy your ROMs into the RAMcard and then modify the card to reset into the monitor.

```
CALL-151
C081 C081
D000<D000.FFFFM
FFFC:59 FF
C080
3D0G
```

If you want to make any additional changes to the monitor (see the articles on improving the F8 ROM in COMPUTIST Nos. 6, 19, and 29), insert them after FFFC:59 FF. After EXECing this file, flip the switch to write-protect and boot your protected disk with a PR#6.

An Owner's Review of...

The Copy Master II

by Andrew Swartz

Applied Technology Laboratories
11926 Santa Monica Blvd.
Los Angeles, CA 90025

Works with Apple][Plus and //e.

About one year ago, I had purchased my Copy Master II copy card in hopes of deprotecting my 'protected software'. The Copy Master version I bought allows the user to back up all 128K single-load software at the push of a button. This package also included screen-dumping software to print text and/or graphics from protected software. At that time, I paid $139.95 for this board. Today, it can be purchased through mail-order companies for about $90 to $100.

The Copying Process

The Copy Master II card definitely will copy all memory resident DOS 3.3 software without any problems except programs that verify the presence of the original disk in the drive. This can still be defeated by modifying the verifying subroutine through the use of the Step/Trace Program supplied with the board (discussed later). The Copy Master is supposed to copy CP/M programs also, though I have never tried this feature.

For programs that are not entirely memory resident, it is possible (but often a pain and very, very difficult) to make a copy of a particular program by segmenting it onto two or more disks. For example, I managed to make a copy of Racter by copying the first stage of the game up to the hi-res title screen with the Copy Master and then bit copying the rest of the game to another disk. Now when I boot Racter and get to the title page, I flip the disk and the game proceeds normally. Other times, disks retrieve small pieces of information constantly and are impossible to copy with the Copy Master. I am sure if you are a computer wizard, you could manage to link the different segmented files created by the copy Master on one disk thus saving disk space. The Copy Master manual contains very little information on using the board to copy multiple access software.

This copy card also comes with conversion software (which you select from the copy program disk menu) that allows you to make copies 'board independent', meaning you won't need to have the Copy Master in your Apple to run your copies. The Step/Trace program allows users to debug and examine a program (even a protected one). With a press of the Copy Master trigger switch, this program will execute instructions in a program one by one. It will display the last four instructions with the values of the accumulator, x-register, y-register, status register, and stack pointer.

The File Writer program is used when the user wants to modify code produced by the copy program. He or she is allowed to change code, then this program will write the code back to the disk. The Pause Program feature is used to pause a program. It is really a useless feature and should have been replaced by an enter-the-monitor feature, which the Copy Master II lacks. (After writing Applied Technology Labs asking why the ability to enter the monitor was left out of the Copy Master II card, they informed me that their new product called the Easy Card, which combines the power of the Copy Master II card with a clock/calendar and other functions, does have the ability to enter the monitor.)

The last two programs on the menu, System Test and System Configure, test your Apple's main and auxiliary memory, the on-board 4K RAM in the Copy Master (where the Copy program is stored), and allows you to change the slot your Copy Master resides in. (The copy Master II board will fit in any slot except the Apple IIe auxiliary slot.)

One of the best features of the Copy Master Card is the screen-dump software. With software text and/or graphics can be sent to a variety of printers with a variety of interface cards. It works extermely well even with protected software. On dumping graphics or graphics with text you can select from emphasized printing, hi-res or double hi-res (on the IIe), inverse graphics or text, and double-sized graphics. I own an Apple Imagewriter with an Apple Super Serial Card, I can say this feature works pretty well. Another nice feature is the ability to save hi-res or double hi-res screens to a disk.

Closing Thoughts

Overall, the Copy Master II is a so-so buy. In my opinion, if you are going to buy a card specifically for copying, look into some other copy cards like the Wildcard II which has the enter-the-monitor feature. I feel that that should be a necessary feature of all NMI copy cards. I don't see how the people at Applied Technology Labs could have forgotten this important feature. However, I must also say that the ability of the Copy Master II to dump graphics and text from protected software makes up somewhat for this loss.

If you are planning to buy a copy card, you should keep in mind the difficulty in backing up multiple access software with copy cards, since they can only retrieve information that's already in the computer's memory. This may make you want to consider buying a decent bit-copy program like Copy II Plus to use in conjunction with a copy card.

Softkey for...

Mouse Calc

by Richard Glasgow

Requirements:
COPYA
Sector Editor

The first thing I do when I buy a new program is try to back it up before I try to use it. When I used COPYA to backup Mouse Calc, I would get the message: "Reboot Mouse Calc." The next thing I did was to watch the boot by taking off my drive cover and looking at the tracks the program uses. I have the tracks marked on the drive as described in the EDD III manual. This speeds up the process of finding the area where the protection is.

On Mouse Calc, I noticed that the last track accessed was track $13. So, I got out my trusty sector editor and started snooping around this track. I started inserting zero bytes in each sector (on the copy disk of course), until I noticed a change in the boot. Sure enough, when I reached sector $09, I did not get the reboot message. Looking further at the code on sector $09, I didn't find much that made any sense. So, I went to sector $08 and there I found some code that looked interesting. Upon examining and experimenting with this code a little, I noticed that if I changed byte $29 from a $07 to a $48, the program would boot just fine.

The code surrounding this byte looked something like this:

```
A8D1- A9 20     LDA #$20
A8D3- 85 06     STA $06
A8D5- A2 00     LDX #$00
A8D7- E8        INX
A8D8- D0 07     BNE $A8E1
A8DA- C6 06     DEC $06
A8DC- D0 03     BNE $A8E1
A8DE- 4C 69 A9  JMP $A969
A8E1- AD 00 03  LDA $0300
```

What my modification does is change the BNE $A8E1 to a BNE $A922.

Step by Step

1) Copy the program using COPYA or any other fast sector by sector copier.

2) Scan the copied disk for the sequence $A2 00 E8 D0 07 C6 06 and when found, change the $07 to a $48. I found this sequence on track $13, sector $08, byte $25.

Keys to success on the...

Franklin Ace

by Kevin D. Miller

Requirements:
Franklin Ace 100 that does not generate all ASCII characters
EPROM burner
2758 or 2716 EPROM

One aspect of the Franklin Ace computer that I enjoy is the upper/lower case and 10-key pad keyboard. As a programmer on a mainframe computer system, and in writing a terminal driver for my Franklin Ace 100 to work via a modem onto the mainframe, this almost-complete keyboard still wasn't enough.

This keyboard does not generate the printing characters backslash, tilde, accent, bar, and the non-printing characters <DEL> ,<Field Separator>, <Group Separator>, <Record Seperator>, and <Unit Seperator>.

After examining the keyboard circuitry, I found that each key, when pressed, addresses an EPROM which in turn outputs the 7 bit ASCII value for that key. All I needed to do was to re-program the EPROM with the missing characters and then I could generate the full ASCII character set.

The EPROM in my keyboard was a 2758, 24 pin, 1K EPROM. I had a spare 2716, 24 pin, 2K EPROM which happens to be pin-compatible. On the adjacent page is a dump of my old 2758 keyboard EPROM with the changed values listed along side. The listing is divided into different combination groups of each of the special select keys, Lock, Control and Shift. You will notice that the hex value given for any particular character does not look familiar. These values are the complement of the ASCII value sent to the computer. For example, the character 'A' in the listing has the value $BE. After Exclusive-ORing the value with $FF, you get the value $41, which is an "A." The changes I made are as follows.

keyboard	old value	new char	ASCII value	EPROM value
ctrl /	$D0	\	$5C	$A3
ctrl '	$D8	`	$60	$9F
ctrl-shift 1	$DE	|	$7C	$83
ctrl -	$D2	~	$7E	$81
> on keypad	$41	<DEL>	$7F	$80
ctrl-shift F	$F9	<FS>	$1C	$E3
ctrl-shift G	$F8	<GS>	$1D	$E2
ctrl-shift R	$ED	<RS>	$1E	$E1
ctrl-shift U	$EA	<US>	$1F	$E0

I also changed my shift keys to be a toggle-case key as on the IBM keyboard, so that with lock off, shift produced uppercase, and with lock on, shift produced lowercase.

With this method, you can change any key to produce any ASCII value. However, since the letter "A" is painted on the top of a key, it may be confusing if it generated anything else but an "A".

For those that may be interested, I have also increased the auto-repeat rate for the keys by replacing one resistor. I do not claim to be an expert in the electronics field, but if any readers have questions about these procedures, their inquiries are welcome.

The Keyboard map EPROM

```
   old eprom listing          key definitions
  -------------------        -----------------
$000: nolock, nocntrl, noshift

FF CD CE FF F6 8E 9E 8C      2  1    tab q  a  s
85 CC CB 9A 8D 88 9B 87   z  3  4  e  r  w  d  x
9C CA C9 8B 86 99 98 89   c  5  6  t  y  f  g  v
9D C8 C7 8A 96 97 95 91   b  7  8  u  i  h  j  n
DF C6 CF 90 8F 94 93 92  spc 9  0  o  p  k  l  m
D3 D2 C2 84 A4 C4 D8 D1   .  -  =  {  [  ;  '  ,
D0 55 6A FF 7C F2 77 FF   /  *  →    brk cr ←
FF 48 47 4B 4A 54 4E 41      7  8  4  5  +  1  >  (keypad)
4F 46 FF FF 49 4D 4C 51   0  9        6  2  3  .  (keypad)
FF FF FF FF FF 6C FF 64                  pau   esc
FF FF FF FF FF FF FF 52                        -  (keypad)
FF FF ...

$100: nolock, nocntrl, SHIFT

FF BF DE FF F6 AE BE AC   The above key pattern
A5 DC DB BA AD A8 BB A7   is repeated for each
BC DA A1 AB A6 B9 B8 A9   select group.
BD D9 D5 AA B6 B7 B5 B1
DF D7 D6 B0 AF B4 B3 B2
C3 A0 D4 82 A2 C5 DD C1
C0 55 6A FF 7C F2 77 FF
FF 48 47 4B 4A 54 4E 41
4F 46 FF FF 49 4D 4C 51
FF FF FF FF FF 6C FF 64
FF FF FF FF FF FF FF 52
FF FF ...

$200: nolock, CONTROL, noshift
                                New eprom values
FF CD CE FF F6 EE FE EC   --  --  --  --  --  --  --  --
E5 CC CB FA ED E8 FB E7
FC CA C9 EB E6 F9 F8 E9
FD C8 C7 EA F6 F7 F5 F1
DF C6 CF F0 EF F4 F3 F2
D3 D2 C2 84 E4 C4 D8 D1   D3 <81> C2  84  E4  C4 <9F> D1
D0 55 6A FF 7C F2 77 FF  <A3> 55  6A  FF  7C  F2  77  FF
FF 48 47 4B 4A 54 4E 41   FF  48  47  4B  4A  54  4E <80>
4F 46 FF FF 49 4D 4C 51   4F  46  FF  FF  49  4D  4C <83>
FF FF FF FF FF 6C FF 64
FF FF FF FF FF FF FF 52   FF  FF  FF  FF  FF  FF  FF <81>
FF FF ...

$300: nolock, CONTROL, SHIFT

FF FF DE FF F6 EE FE EC   FF  FF <83> FF  F6  EE  FE  EC
E5 DC DB FA ED E8 FB E7   E5  DC  DB  FA <E1> E8  FB  E7
FC DA A1 EB E6 F9 F8 E9   FC  DA <E1> EB  E6 <E3><E2> E9
FD D9 D5 EA F6 F7 F5 F1   FD  D9  D5 <E0> F6  F7  F5  F1
DF D7 D6 F0 EF F4 F3 F2
C3 A0 D4 82 E2 C5 DD C1   C3 <E0> D4  82  E2  C5  DD  C1
C0 55 6A FF 7C F2 77 FF
FF 48 47 4B 4A 54 4E 41   FF  48  47  4B  4A  54  4E <80>
4F 46 FF FF 49 4D 4C 51   4F  46  FF  FF  49  4D  4C <83>
FF FF FF FF FF 6C FF 64
FF FF FF FF FF FF FF 52
FF FF ...

$400: LOCK, nocntrl, noshift

FF CD CE FF F6 AE BE AC
A5 CC CB BA AD A8 BB A7
BC CA C9 AB A6 B9 B8 A9
BD C8 C7 AA B6 B7 B5 B1
DF C6 CF B0 AF B4 B3 B2
D3 D2 C2 84 A4 C4 D8 D1
D0 55 6A FF 7C F2 77 FF
FF 48 47 4B 4A 54 4E 41   FF  48  47  4B  4A  54  4E <80>
4F 46 FF FF 49 4D 4C 51
FF FF FF FF FF 6C FF 64
FF FF FF FF FF FF FF 52
FF FF ...

$500: LOCK, nocntrl, SHIFT

FF BF DE FF F6 AE BE AC   FF  BF  DE  FF  F6 <8E><9E><8C>
A5 DC DB BA AD A8 BB A7  <85> DC  DB <9A><8D><88><9B><87>
BC DA A1 AB A6 B9 B8 A9  <9C> DA  A1 <8B><86><99><98><89>
BD D9 D5 AA B6 B7 B5 B1  <9D> D9  D5 <8A><96><97><95><91>
DF D7 D6 B0 AF B4 B3 B2   DF  D7  D6 <90><8F><94><93><92>
C3 A0 D4 82 A2 C5 DD C1
C0 55 6A FF 7C F2 77 FF
FF 48 47 4B 4A 54 4E 41   FF  48  47  4B  4A  54  4E <80>
4F 46 FF FF 49 4D 4C 51
FF FF FF FF FF 6C FF 64
FF FF FF FF FF FF FF 52
FF FF ...

$600: LOCK, CONTROL, noshift

FF CD CE FF F6 EE FE EC
E5 CC CB FA ED E8 FB E7
FC CA C9 EB E6 F9 F8 E9
FD C8 C7 EA F6 F7 F5 F1
DF C6 CF F0 EF F4 F3 F2
D3 D2 C2 84 E4 C4 D8 D1   D3 <81> C2  84  E4  C4 <9F> D1
D0 55 6A FF 7C F2 77 FF  <A3> 55  6A  FF  7C  F2  77  FF
FF 48 47 4B 4A 54 4E 41   FF  48  47  4B  4A  54  4E <80>
4F 46 FF FF 49 4D 4C 51   4F  46  FF  FF  49  4D  4C <83>
FF FF FF FF FF 6C FF 64
FF FF FF FF FF FF FF 52   FF  FF  FF  FF  FF  FF  FF <81>
FF FF ...

$700: LOCK, CONTROL, SHIFT

FF FF DE FF F6 EE FE EC   FF  FF <83> FF  F6  EE  FE  EC
E5 DC DB FA ED E8 FB E7   E5  DC  DB  FA <E1> E8  FB  E7
FC DA A1 EB E6 F9 F8 E9   FC  DA <E1> EB  E6 <E3><E2> E9
FD D9 D5 EA F6 F7 F5 F1   FD  D9  D5 <E0> F6  F7  F5  F1
DF D7 D6 F0 EF F4 F3 F2
C3 A0 D4 82 E2 C5 DD C1   C3 <E0> D4  82  E2  C5  DD  C1
C0 55 6A FF 7C F2 77 FF
FF 48 47 4B 4A 54 4E 41   FF  48  47  4B  4A  54  4E <80>
4F 46 FF FF 49 4D 4C 51   4F  46  FF  FF  49  4D  4C <83>
.....
```

Modified F8 ROMs On the Apple ///

by Eric Deslauriers

Requirements:
An Apple III
A copy of the Emulation disk
A normal DOS 3.3 Disk

Have you ever been disappointed by seeing a "flip Integer switch and hit Reset" message in the midst of a softkey or had no way of breaking out of a protected program without the Ctrl-Reset (which does little good if you are trying to crack programs)? Well, now you can stand up and cheer because the software modifications in this article will allow you to use the Modified F8 ROM written by Ernie Young in COMPUTIST #6, pg 14. (That issue is no longer available, but the hexdump is supplied here.)

This modified ROM allows the user to select between a normal autostart reset, an old monitor style reset, or one that saves memory locations $0000-$0900 and puts them in locations $2000-$2900. This option will also save the stack pointer at location $2091. The new code is tucked into the F8 area by eliminating the cassette I/O.

The best part about this procedure is that you don't have to mess around with burning EPROMs (Eraseable Programmable ROMs). And the cost of this is a little time and a disk.

This procedure was born of a necessity to be able to hit Reset and be dropped into the monitor. After getting very frustrated at the Lode Runner softkey (COMPUTIST No. 22, p11), I decided to try to have my Emulation disk tell my Apple III that it had a different "ROM". This procedure was accompanied by much swearing and a lot of learning. But, as a novice at machine language and a lot of everything else, I am very proud to have accomplished this. And, sometimes, I am very happy that I decided to do this.

The Procedure

1) Make a duplicate copy of your Emulation Disk and hide the original.

2) Insert your copy of the Emulation Disk into the internal drive and turn your computer on.

3) Boot up with Applesoft BASIC. For those of you unfamiliar as to how to get into Applesoft, boot the Emulation disk (insert it into the internal drive and press Ctrl-Reset or turn your Apple on). When presented with the menu, press ESCAPE. At the top of the screen, an arrow will be pointing at the choice labeled:

```
LANGUAGE: APPLESOFT INTEGER
```

Using the right or left arrow, move the light bar onto the choice labeled APPLESOFT. Insert your DOS 3.3 disk (You must have a disk with normal DOS 3.3 because one of the accompanying programs, SREAD/SWRITE, use locations specific to DOS 3.3.) and press RETURN to boot it.

3) Exit your HELLO program, clear memory, and hop into the monitor by typing:

FP
CALL -151

4) Then type in the hexdump of SREAD/SWRITE, which can be found in listing 1. This program by Bryan Farlow and Robert Knowles appeared in COMPUTIST No. 24, p 16. It is used to load a sector into memory and to write it back onto disk. But remember this; SREAD/SWRITE will ONLY read and write from and to the disk in drive one. For more information, you should refer to that article. Save SREAD/SWRITE with:

BSAVE SREAD/SWRITE,A$300,L$A4

5) Exit the monitor and clear memory by typing

FP

6) Type in the BASIC program, TRACKMOVER, found in listing 2. This program will use the SREAD/SWRITE program to load one of the disk's tracks into the memory at the location we tell it to load into. Save it with

SAVE TRACKMOVER

7) Clear the program and enter the monitor again.

FP
CALL -151

8) Type in the modified F8 ROM hexdump found in listing 3. Save it with

BSAVE PSUEDO-ROM,A$2ECD,L$5A

9) Hop back into APPLESOFT with

10) The Emulation disk has two tracks that have to be modified for the ROM to be changed. Track 5 contains the "ROM" for INTEGER BASIC and track 9 contains the "ROM" for APPLESOFT. You may or may not want to change the INTEGER "ROM". I changed them both for curiosity and for conformity.

There is a confusing aspect to the memory for the "ROM". Apple decided not to have the ROMs start at the same sector (different tracks) on the disk. This means that, when the tracks are loaded into memory, the INTEGER "ROM" starts at x9CD while the APPLESOFT "ROM" starts at x4CD ('x' being the page the track was loaded into.). This makes the changes a little more confusing but not too much so.

11) Run TRACKMOVER and then insert your Emulation disk in drive 1 (be sure it's in drive 1!). Now we will load track 5 at $5000 by typing what is indicated after the prompts below.

RUN TRACKMOVER

(You DID insert your Emulation disk, right?)

```
READ/WRITE (R/W): R
SREAD TRACK : 5
HEX LOCATON: 5000
```

Answer Y to SREAD ANOTHER TRACK? and load track 9 at $4000 by typing

Y
9
4000

12) This time type **N** and you will be dropped into the monitor.

13) Use the monitor move command to copy the modified ROM hexdump (that we previously stored in memory) onto the track 5 and track 9 images.

59CD<2ECD.2F27M
44CD<2ECD.2F27M

14) Now the Reset vectors have to point to the new code so that when Reset (not Ctrl-Reset as the computer will reboot and become an Apple III) is pressed, the program is executed. Unlike the Apple II owners, though, we had better not change the NMI (Non Maskable Interrupt) vectors; because if we do, every time we press Reset, the computer goes off and beeps at you just about forever.

Change the RESET vectors with

5AFA:CD FE
45FA:CD FE

15) Exit the monitor with Ctrl-C. If you try anything else (such as E003G or 3D0G), the computer will let you hit RETURN several times and then go into a frenzied fit. Furthermore, any command you type in will be answered with ?SYNTAX ERROR. I haven't found out why this occurs yet. Anyone have an answer?

16) Run TRACKMOVER and save track 5 from $5000 by typing

RUN
W
5
5000

Answer Y and save track 9 from $4000:

Y
5
4000

17) Voila!!! Now, every time you need to RESET out of a protected program, just boot your altered Emulation disk and go to it. Who needs chips and burners when ya got an Apple III?!

A Short Tutorial

As the proud owner of your new F8 ROM, it is only fitting that you learn how to use it. Luckily, the new F8 ROM isn't very hard to learn how to use.

To start learning, boot any Apple II disk. When the disk finishes booting, press Reset. The system will "hang." Actually, it is waiting for you to press one of three keys. These keys are the Return key, the "-" key, and the ":" key.

For an autostart ROM Reset (the one you usually get when you hit Reset), press the Return key. Look familiar?

This time, we'll try for an old style Reset. This Reset will just drop you into the monitor. Press Reset again and then press the "-" key. Did the "*" prompt show up at the bottom of the screen? This will let you snoop through the code. No more reboots with this baby!

Now for the Saver function. Press RESET once more. This time press the ":" (shift ";") key. Did the screen clear and the "*" prompt show up at the bottom of the screen?

What this just did was to take whatever was in memory locations $0000-$0900 and store the values in locations $2000-$2900. Then it saved the stack pointer at $2901.

That's all that's needed to know to use your new ROM. Enjoy your cracking!

Sread/Swrite (Issue No. 24)

```
0300: A0 04 B9 48 03 99 93 A8   $193C
0308: 88 10 F7 A0 05 B9 4D 03   $3C10
0310: 99 C6 A8 88 10 F7 A9 D4   $CBF8
0318: 8D 41 A9 A0 03 B9 53 03   $2EF7
0320: 99 5D A9 88 10 F7 A9 40   $9E0F
0328: 8D 11 A9 8D 25 A9 A9 71   $C6CD
0330: 8D 12 A9 8D 26 A9 A0 02   $B7E1
0338: B9 57 03 99 F0 A4 B9 5A   $C8D4
0340: 03 99 98 A2 88 10 F1 60   $683F
0348: 53 52 45 41 C4 53 57 52   $347F

0350: 49 54 C5 00 00 0F 00 4C   $81BA
0358: 65 03 4C 5D 03 A9 02 8D   $6D85
0360: 9B 03 4C 6A 03 A9 01 8D   $CAC6
0368: 9B 03 AD 66 AA 8D 93 03   $FC54
0370: AD 6A AA 8D 94 03 AD 72   $279F
0378: AA 8D 97 03 AD 73 AA 8D   $0A53
0380: 98 03 A9 03 A0 8F 20 D9   $5265
0388: 03 A9 06 8D 6A AA 60 01   $B00B
0390: 60 01 00 00 00 A0 03 00   $D64A
0398: 00 00 00 00 00 00 60 01   $C70A
03A0: 00 01 EF D8               $8F9B
```

F8 ROM Patch (Issue No. 6)

```
2ECD: 2C 00 C0                  $0E8B
2ED0: 10 FB 8D 10 C0 AD 00 C0   $B6A9
2ED8: C9 2D F0 7D B0 03 4C 62   $1B98
2EE0: FA BA 8E 01 29 A0 00 B9   $D15A
2EE8: 00 00 99 00 20 B9 00 01   $9B59
2EF0: 99 00 21 4C FD FE 20 00   $1FE5
2EF8: FE 68 68 D0 6C C8 D0 E7   $C5C6
2F00: 84 3C 84 42 84 3E A9 09   $2A77
2F08: 85 3F A9 02 85 3D A9 22   $9E7A
2F10: 85 43 20 2C FE 20 2F FB   $F0AC
2F18: 20 58 FC 4C 59 FF 60      $66E7
```

TrackMover

```
10 REM ************************
20 REM         TRACKMOVER
30 REM   BY ERIC DESLAURIERS
40 REM COPYRIGHT 1986 COMPUTIST
50 REM ************************
60 HOME
70 IF PEEK (768 ) < > 160 THEN GOSUB 360
80 FLASH : VTAB 2 : PRINT "BE^ SURE^ THE^
    EMULATION^ DISK^ IS^ IN^ DRIVE^ 1" :
    NORMAL
90 PRINT : PRINT "READ^ OR^ WRITE^ TRACK?^
    (R/W)" ;
100 GET A$
110 IF A$ = "R" THEN COM$ = "SREAD" : GOTO 140
120 IF A$ = "W" THEN COM$ = "SWRITE" : GOTO
    140
130 GOTO 100
140 PRINT : VTAB 8 : PRINT COM$; "^ TRACK:^ "
    ; : INPUT "" ;HEX$ : GOSUB 270
150 TK = DEC : IF TK > 34 THEN PRINT "NO^
    TRACKS^ ABOVE^ $22" : GOTO 140
160 VTAB 10 : PRINT "HEX^ LOCATION:^ " ; :
    INPUT "" ;HEX$ : GOSUB 270
170 LOC = DEC : IF LOC < 8192 THEN PRINT "NO^
    NUMBERS^ LOWER^ THAN^ $2000^ PLEASE" :
    GOTO 160
180 IF LOC > 38400 THEN PRINT "NO^ NUMBERS^
    HIGHER^ THAN^ $9600^ PLEASE" : GOTO
    160
190 FOR SEC = 15 TO 0 STEP - 1
200 PRINT CHR$ (4 );COM$; "^ T" ;TK; ",S"
    ;SEC; ",A" ;LOC
210 LOC = LOC + 256
220 NEXT SEC
230 VTAB 12 : PRINT COM$; "^ ANOTHER^ TRACK?^
    (Y/N)" ; : GET A$ : IF A$ = "Y" THEN HOME
    : GOTO 140
240 IF COM$ = "SREAD" THEN CALL - 151
250 END
260 REM
    * * * CHANGE HEX$ FROM HEX TO DECIM-
    AL (DEC )
270 HX$ = "0123456789ABCDEF" :DEC = 0
280 FOR I = 1 TO LEN (HEX$ )
290 C = ASC ( MID$ (HEX$ , I ,1 ) ) - 48
300 IF C > 9 THEN C = C - 7
310 IF C < 0 OR C > 15 THEN PRINT "YOU^ HAVE^
    MISTYPED^ THIS^ NUMBER." : GOTO 140
320 DEC = DEC + C + 15 * DEC
330 NEXT I
340 RETURN
350 REM  * * * BRUN SREAD/SWRITE
360 PRINT "INSERT^ THE^ DISK^ WITH^
    SREAD/SWRITE" : INPUT "AND^ PRESS^
    RETURN." ;X$
370 PRINT CHR$ (4 ); "BRUN^ SREAD/SWRITE"
380 VTAB 5 : PRINT "NOW^ INSERT^ YOUR^
    EMULATION^ DISK^ INTO" : INPUT "DRIVE^
    1^ AND^ PRESS^ RETURN" ;X$
390 HOME : PRINT : RETURN
```

controller checksums

```
10   - $BADD     210  - $05C4
20   - $9B13     220  - $E0C1
30   - $4D3B     230  - $399B
40   - $AD92     240  - $944F
50   - $C899     250  - $BB93
60   - $DAF7     260  - $A7A3
70   - $6AB4     270  - $0DEC
80   - $9555     280  - $B46A
90   - $36EF     290  - $BF67
100  - $55DC     300  - $680E
110  - $3657     310  - $9813
120  - $54CD     320  - $2F0F
130  - $EB5C     330  - $F3C7
140  - $0E6E     340  - $A06A
150  - $E826     350  - $0ABF
160  - $6E49     360  - $B50A
170  - $4E6E     370  - $A82D
180  - $2E58     380  - $F888
190  - $199B     390  - $C032
200  - $AA4D
```

softkey for...

Sands of Egypt

by Steve Ellis

Datasoft

Requirements:
At least 48K
A blank disk
Sands of Egypt disk
A sector editor

Sands of Egypt is an interesting and difficult adventure game that has some pretty good animated graphics. Its protection technique seems unusual at first glance, but after a little investigation it can be found that all it consists of is a nibble count.

The disk contains useful information from tracks $0-$11, and after that the rest of the disk is filled with garbage. The first step I made in deprotecting it was to copy the first $12 tracks with the Manual Sector Copy on Copy II Plus. When I booted this copy, it loaded the hi-res title page then hung while accessing the disk. I assumed it was performing a nibble count, so I NMI'd the program with my Wildcard and disassembled the program code where it had been executing. Sure enough, right there was a long sequence of LDA $C08C,X instructions followed by CMPs to various values.

Tracing backwards from there, I found that the nibble count routine started at $911D in memory. On the original disk this routine is erased after the nibble count is successful, so the only way to see it is to boot the copy. I wrote down the first five bytes of the routine and searched the disk for them. Unfortunately, they could not be found anywhere. This led me to believe that Datasoft was encrypting the code on the disk, then decrypting it after it was read into memory. I examined the memory on page $91 (which was the only page that contained any suspicious code), and found that bytes $95-$FF were all $A0. Then I started examining sectors, starting with track 0, sector 0, looking for a similar pattern - that is, a sector with the same value repeated in bytes $95-$FF. I found what I was looking for on track 0, sector $E. This sector contained what seemed like garbage, but bytes $95 through $FF were all the same value ($75), which followed the pattern of page $91 in memory.

To double check, I disassembled page $91 in memory and found that the encrypting routine was at the beginning of that page, and that the data was encrypted by EORing each byte from $911D to $91FF with $D5. This can be verified using the Senior Prom sector editor. Read in track 0, sector $E and exclusive-or the buffer with $D5. Disassembly will show the nibble count code starting at byte $1D. So to disable the routine that did the nibble count, all that was necessary was to EOR a $60 (for RTS) with $D5 and write the resulting byte to disk at byte $1D of track 0, sector $E. Doing this worked fine, except that the disk drive did not turn off once the game started. So I encrypted the values for LDA $C088,X (to turn off the drive), and wrote those to the disk at byte $1D, and followed them with the RTS byte. This produced a working copy of Sands of Egypt.

Extra Room

If you CATALOG the copy (or the original, for that matter) you will see one program called HELLO. This is a dummy program which cannot be accessed. But it shows that the disk has a normal VTOC and normal CATALOG sectors. For this reason, the blank tracks on the disk (tracks $12-$22) can be used for files. All that is necessary is to fix a few CATALOG sectors, and to fix the VTOC to show tracks $0-$11 as being full and tracks $12-$22 as free. The procedure is shown below. Note that this step is not necessary for the softkey to work; it is just an added feature.

Step By Step

1) Copy the original disk (tracks $0-$11 only) using any program that will copy individual tracks. If you do not have such a program, COPYA can be modified as follows:

```
RUN COPYA
[C]
CALL-151
302:12 N 35F:12
[C]
DEL 70
RUN
```

2) Using a sector editor, write the following bytes to the copy:

Track 0, Sector $E:

Byte#	From	To
$1D	$70	$68
$1E	$2B	$5D
$1F	$58	$15
$20	$41	$B5

3) The softkey is complete at this point. If you want to free up tracks $12-$22:

a) DELETE HELLO from the Sands of Egypt disk.

b) With a sector editor, read in track $11, sector $E. It should contain random garbage. Zero out the entire sector buffer, and change bytes 1 and 2 to $11, $0D. Write the sector back to disk. The same should be done for sectors $D,$C,$B,$7,$6,$5,$4, and $3. On each sector, change byte 2 to the number of the sector minus one. In other words, sector $B, bytes 1 and 2 would read $11, $0A, and so on.

c) Read in the VTOC (track $11, sector 0). Zero out the buffer from bytes $38-$FF. At byte $80, write 2 $FF's. Skip to byte $84 and write two more $FF's. Continue this pattern (write 2 $FF's, skip 2 bytes, write 2 $FF's), stopping at byte $C4. That is, bytes $C2 through $FF should all be zero. Write this modified VTOC back to disk.

d) Now you can use the disk for saving files. You will have 272 free sectors to use.

revisited softkey for...

Number Farm

by Scott McCandless

DLM, Inc.
1 DLM Park
Allen TX 75002

Requirements:
Number Farm diskette
COPYA
A sector editor

I would like to pass on to you a softkey I have made using an interesting patch to DOS for slowing the boot process of a disk. This change can slow down the boot of a disk by 4 times or, in comparison: 14 seconds for the normal disk compared to 63 seconds for the patched disk. The disk I am referring to is one of the lates packages from DLM & Neosoft, "Number Farms." The change I made allowed me to witness the boot at a pace that is easier for me to watch and decipher.

In the past, DLM only protected their disks by altering the data and address epilogues. Well, now not only have they abolished that worthless idea, but they have decided to include a nibble count onto a disk that is COPYAable: "Learning With Fuzzywomps" haunts me.

The Protection

Ever since my mother has purchased educational programs for herself, she has always wanted me to make reliable backups, and write simply bit copies usually cannot handle the newest protections. So, I decided to get into the ever increasing world of deprotection and liberation of my original software packages. Now, this disk posed a problem to me because it had a nibble count farther into the boot than I preferred to boot-code trace. And not only that, the disk does not have a normal catalog. The VTOC is located on track $11 sector 0, while the actual catalog was located on track 3, which I could not get a successful listing without capturing their DOS and executing an "A56EG" (directly calling the catalog) from the monitor. At this point I decided that Demuffin Plus was not going to do it, so I decided to COPYA the disk. After doing so, the disk would continually reboot. So, the next step was to search for a re-boot routine.

Locating the Protection

As everyone should know, looking for the JMP $C600 it the most likely place to substitute a jump to the monitor (JMP $FF59). Well, upon doing that and booting the disk, it once again re-booted without jumping to the monitor. After that unsuccessful try, I was ready to try anything. As I was looking for some disk read routines (LDA $C08C,X) I came across the instruction LDX #$11 on track $0 sector $4. After changing the instruction to LDX #$00, I rebooted the disk. Upon booting the disk, I noticed how slow it loaded and I thought to myself, "this is great!" After watching the disk boot, I noticed that it loaded tracks 0,2,1,0,11,3, then went to track 6, where it then moved the drive cam to track 0 and rebooted. If, however, the nibble count passed, the disk would continue and load the rest of the program. After this little view of the boot, I came to the conclusion that the routine must be on track 6. When I looked at track 6 for some funny routine, I came upon the following code on track 6, sector $D:

```
xx0E: A9 00     LDA #$00    track to count
xx10: 85 FC     STA $FC
xx12: A2 03     LDA #$01
xx14: BC 49 90  LDY $9049,X
xx17: 91 FA     STA ($FA),Y
xx1A: CA        DEX
xx1C: 10 F8     BPL $xx14
xx1D: 48        PHA
xx1E: 20 3C 90  JSR $903C
xx21: A0 01     LDY #$01
xx23: B1 FA     LDA ($FA),Y
xx25: AA        TAX
xx26: 30 58 90  JSR $9058  call count
xx29: 68        PLA
xx2A: 68        PLA
xx2B: A0 00     LDY #$00
xx2D: 84 FE     STY $FE
xx2F: B9 4D 90  LDA $904D,Y
xx32: 09 80     ORA #$80
xx34: 20 39 90  JSR $9039
xx37: A4 FE     LDY $FE
xx39: C8        INY
xx3A: 4C 29 90  JMP $9029

.
.
.
xx46: A9 00     LDA #$00
xx48: 85 48     STA $48
xx4A: B0 5D     BCS $xxA9
xx4C: 60        RTS
...if everything is OK, run hello
```

Now, for an explanation of the above code. Bytes $0E through $1A do a memory move and set up the nibble count parameters (changing byte $0F will make the drive cam seek a different track for the nibble count). Bytes $1C through $26 set up the final parameters and call the nibble count routine. The rest is pretty much clear; if the count fails then the hello file is not run, and the program jumps to a reboot routine.

Step by Step

1) Run COPYA and copy the disk. No modifications are needed.

2) Pull out your sector editor and read in track $06 sector $D. Starting at byte $26, change the bytes 20 58 90 (JSR $9058) to EA EA EA (three NOPs).

3) Write the sector back to the disk and boot it up!

Now, if you would like to see the slow boot process before you take out the copy protection, run your sector editor and do the following:

1) Read track $0, sector $04 from your newly copied backup.

2) Change byte $01 from $11 to $00.

3) Write the sector back to your disk.

4) Remove the case from your disk drive, if possible, and watch the disk boot real slooooooooow...

5) Watch it access track $6 then go to track $0 for the nibble count.

6) Don't forget to undo the change after you're through having fun!

For your information, the patch shown here changes the timing loop (at $B9FD-BA10) used to wait between steps of the drive arm. $11 is the usual value. The patch is effective for just about any DOS disk. Just search for $A2 11 CA (LDX #$11, DEX) and change the $11 to either $00, $11, or $22. Those are the only valid bytes I have been able to change them to that have any effect at all on the speed of the boot. Changing the value to $00 will slow the boot down greatly, while changing it to $22 will slow it down some, but will also make the disk run a lot quicter, i.e. SSI's Computer Quarterback boots whisper quiet. I have found that changing the $11 to any other value makes the drive motor make funny groaning noises as it tries to seek from track to track.

☆ AGENT U.S.A. ☆

by Bill Johnson

Scholastic, Inc.
2931 East McCarty St.
P.O. Box 7502
Jefferson City, MO 65102

Requirements:
Agent U.S.A.
64K or more
COMPUTIST No. 19
48K slave disk with no hello program

Agent U.S.A. is an arcade style game in an educational package. You, as agent U.S.A., are to destroy the fuzzbomb before the entire population of the U.S. is turned into fuzzbodies. It is designed to entertain while teaching U.S. geography, math skills, and problem solving.

The program is difficult to backup with bit copiers since it accesses tracks 1, 1.5, and 2. I have been successful using the following parameters with E.D.D. III and would recommend using a backup for this softkey. Copy tracks 0,1,1.75 mode 5 then tracks 3-A mode 1. Now, on with the softkey.

Since the program loads in completely at bootup, I decided to convert it to a normal DOS 3.3 file. To do this we must save any memory that is lost when booting a standard DOS disk in a safe place. The hi-res pages ($2000-$5FFF) are available for our temporary use and they provide more than enough space for our needs.

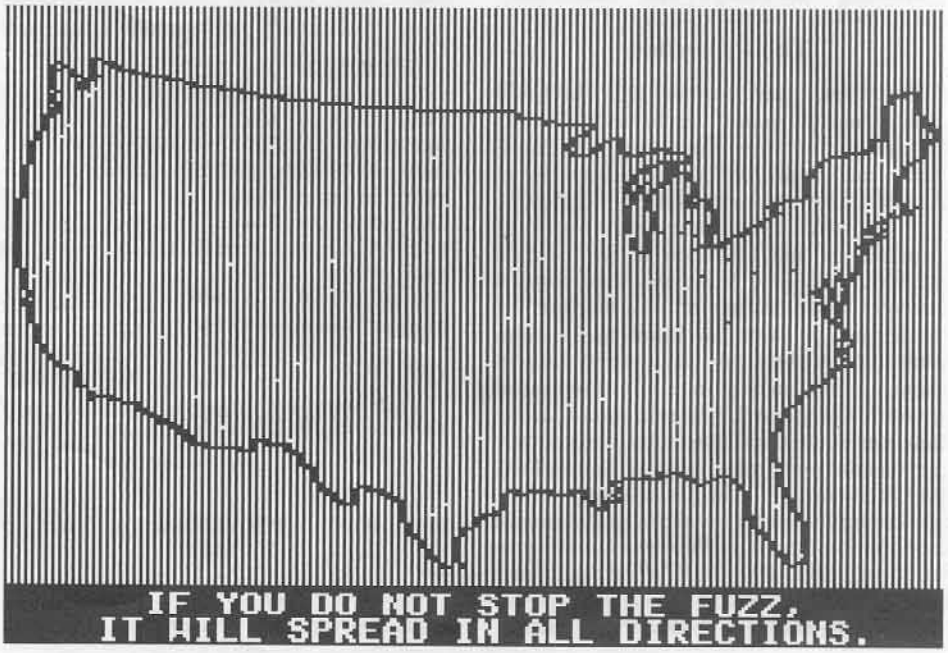
The first problem is how to save memory from $0-$7FF since much of this memory is altered when we reset. The ideal solution would be Earl Taylor's F8 ROM, but I have no desire to modify my motherboard to accept EPROMs. With a little modification, we can use a RAM version of this F8 ROM which will work just as well in many instances.

To make your RAM/ROM, follow the procedures outlined on page 18 of COMPUTIST No. 19 through step #3. Before saving the code, we will add a short routine to move it to the proper destination in our RAM card and turn it on.

```
27D7-   2C 81 C0    BIT   $C081
27DA-   2C 81 C0    BIT   $C081
27DD-   A9 00       LDA   #$00
27DF-   85 42       STA   $42
27E1-   A9 F8       LDA   #$F8
27E3-   85 43       STA   $43
27E5-   A9 00       LDA   #$00
27E7-   85 3C       STA   $3C
27E9-   A9 28       LDA   #$28
27EB-   85 3D       STA   $3D
27ED-   A9 FF       LDA   #$FF
27EF-   85 3E       STA   $3E
27F1-   A9 2F       LDA   #$2F
27F3-   85 3F       STA   $3F
27F5-   A0 00       LDY   #$00
27F7-   20 2C FE    JSR   $FE2C
27FA-   2C 80 C0    BIT   $C080
27FD-   4C 69 FF    JMP   $FF69
```

Save your new RAM/ROM to disk.

BSAVE RAM/ROM,A$27D7,L$829



Now, we can install the new features of the altered ROM into our RAM card by typing **BRUN RAM/ROM**.

In order to keep the program from turning off the RAM card we must alter Boot 1 (Track 0, Sector 0) slightly. Move the disk controller ROM to $8600 and modify it to exit to the monitor after loading Boot 1.

```
8600<C600.C6FFM
86F8:4C 59 FF
8600G
C0E8
```

If you list the code at $801 you will see that the motherboard ROM is turned on at address $81A. We need to NOP this command to keep our RAM F8 in control. Next, relocate the modified code to $86F8 and execute it.

```
81A:EA EA EA
86F8<801.8FFM
8600G
```

The program should load in completely. After the drive stops, press Ctrl-Reset followed by S. You will be returned to the monitor after $0-$8FF has been saved to $2000-$28FF. We must now move $9600-$BFFF to a safe area before booting our 48K slave. $3600 is a convenient location.

```
Ctrl-Reset S
3600<9600.BFFFM
```

(Insert 48K slave and boot it)

```
6 P
```

The last step prior to saving the program is to add a routine to restore memory and jump to the start of the program. I prefer to use the monitor move routine except where the zero page is involved, so this routine gives two different examples of moving memory. (See the box for the listing).

Before we save the file we need to place a jump to our restore routine at the beginning of the file and modify DOS to allow BSAVEing a file this large

```
8FD:4C 00 29
A964:FF
BSAVE AGENT USA, A$8FD,L$8D03
```

The Cookbook

1) Create and install the RAM version of the modified F8 from COMPUTIST No. 19.

```
BRUN RAM/ROM
```

2) Alter Boot1 to keep the RAM card selected.

```
8600<C600.C6FFM
C6F8:4C 59 FF
8600G
C0E8
81A:EA EA EA
86F8<801.8FFM
```

3) Execute it.

```
8600G
```

4) After the drive stops, reset and save $0 - $8FF using the 'modified RAM'.

```
Ctrl-Reset S
```

5) Move the DOS area of memory to a safe place

```
3600<9600.BFFFM
```

6) Insert 48K slave with no hello program and boot it.

```
6 P
CALL -151
```

7) Add a memory restore routine

```
2900:A9 00 85 42 A9 96 85 43
2908:A9 00 85 3C A9 36 85 3D
2910:A9 FF 85 3E A9 5F 85 3F
2918:A0 00 20 2C FE A2 00 BD
2920:00 20 95 00 BD 00 21 9D
2928:00 01 BD 00 22 9D 00 02
2930:BD 00 23 9D 00 03 BD 00
2938:24 9D 00 04 BD 00 25 9D
2940:00 05 BD 00 26 9D 00 06
2948:BD 00 27 9D 00 07 BD 00
2950:28 9D 00 08 E8 D0 C8 4C
2958:37 08
```

8) Place a jump to the restore routine at the start of the file.

```
8FD:4C 00 29
```

9) Alter DOS to allow the BSAVEing of large files and save the file.

```
A964:FF
BSAVE AGENT USA,A$8FD,L$8D03
```

You now have a deprotected Agent U.S.A. which I recommend FIDing to a disk with a fast DOS. Now get out there and destroy that fuzzbomb.

Listing of memory move routine for this softkey

```
*--------------------------------------------------------*
    MOVE $3600 - $5FFF TO $9600 - $BFFF
*--------------------------------------------------------*
2900-  A9 00     LDA  #$00
2902-  85 42     STA  $42   destination Lo
2904-  A9 96     LDA  #$96
2906-  85 43     STA  $43   destination Hi
2908-  A9 00     LDA  #$00
290A-  85 3C     STA  $3C   Memmove beginning Lo
290C-  A9 36     LDA  #$36
290E-  85 3D     STA  $3D   Memmove beginning Hi
2910-  A9 FF     LDA  #$FF
2912-  85 3E     STA  $3E   Memmove end Lo
2914-  A9 5F     LDA  #$5F
2916-  85 3F     STA  $3F   Memmove end Hi
2918-  A0 00     LDY  #$00  Y register must be zero prior
291A-  20 2C FE  JSR  $FE2C to calling Memmove
*--------------------------------------------------------*
            Move $2000 - $28FF to $0
*--------------------------------------------------------*
291D-  A2 00     LDX  #$00
291F-  BD 00 20  LDA  $2000,X
2922-  95 00     STA  $00,X
2924-  BD 00 21  LDA  $2100,X
2927-  9D 00 01  STA  $0100,X
292A-  BD 00 22  LDA  $2200,X
292D-  9D 00 02  STA  $0200,X
2930-  BD 00 23  LDA  $2300,X
2933-  9D 00 03  STA  $0300,X
2936-  BD 00 24  LDA  $2400,X
2939-  9D 00 04  STA  $0400,X
293C-  BD 00 25  LDA  $2500,X
293F-  9D 00 05  STA  $0500,X
2942-  BD 00 26  LDA  $2600,X
2945-  9D 00 06  STA  $0600,X
2948-  BD 00 27  LDA  $2700,X
294B-  9D 00 07  STA  $0700,X
294E-  BD 00 28  LDA  $2800,X
2951-  9D 00 08  STA  $0800,X
2954-  E8        INX
2955-  D0 C8     BNE  $291F
2957-  4C 37 08  JMP  $0837   jump to program start
```

Wavy Navy

by Rich Etarip

Sirius Software Inc.
Sacramento, CA

Requirements:
Apple][Plus
2 blank initialized disks with no HELLO program
A sector editor
Wavy Navy by Sirius Software

Wavy Navy is a challenging and quite addictive overseas 'Space Invaders' type game from Sirius Software. When I first booted up Wavy Navy, I thought it was going to be a simple 'Boot Code Trace and save the program' softkey because once the game is in memory, the disk is not accessed again. However, taking a second look (possibly a third and a fourth too) I found that I had spoken too soon. I suspected that there was something fishy about this protection because the game is a single load program and the disk is easy to boot code trace.

After examining the program in memory, I discovered what this protection was built around. Not on the disk, but rather on the usage of memory. The program lives from $800 to $BAFF with no spare memory in between. This could cause a problem in trying to convert the program into a DOS 3.3 file. I ran into this same problem with another game about a year ago so this time I was prepared and I knew what I had to do to get this game on normal DOS. There are several different ways this can be approached but I chose the method that I felt would be easiest to do, explain, and understand. What this softkey will involve is saving most of the game as a normal DOS 3.3 file and storing the rest directly to the disk, using the RWTS to read it in. Another problem is encountered here. The program occupies memory from $800 to $BAFF and the RWTS uses $B800 thru $BFFF. I thought about this for a while and figured that the best way to do it is to read 4 pages of memory into the text page which is from $400 to $7FF. That way there will be just enough room for the RWTS.

The program will be saved and loaded as follows: memory from $800 to $8FFF will be saved as a DOS 3.3 file; memory from $9000 to $BAFF will be written directly to tracks $20 through $22 on the disk, then read in from the first file using the RWTS. The last 4 pages,$B7 thru $BA, will be read into the text page at $400 and then moved to $B700. We must also modify the track bit map for tracks $20 through $22 so they are not overwritten by any other file you may wish to save on the backup Wavy Navy disk.

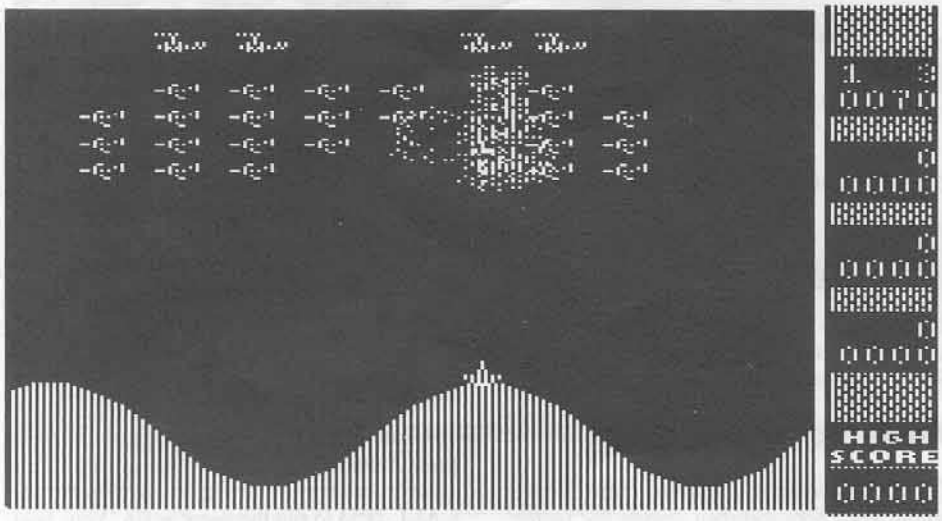
Now that the deprotection process has been explained, we are ready to begin the dissection.

1) The first step is entering the monitor to begin the Boot Code Trace.

```
CALL -151
```

2) Now we must move the boot program from the disk controller card (at $C600) down to RAM so we can modify it to read in the first boot stage and return control to us.

```
9600<C600.C6FFM
```



The Loader

Hexdump 1 uses the RWTS to load section 2 of the game directly from the disk. If you are unfamiliar with the RWTS it is documented on pages 94-98 of your DOS manual. Hexdump 1 has four main sections to it.

Here is an explanation of what it is doing:

● Section 1 copies the entire routine (at $9000) down to page 3 ($300) because the routine loads section 2 of Wavy Navy into $9000

```
9000-  LDX #$00     --the X register is our counter
9002-  LDA $9000,X  --load a byte from memory location $9000+X
9005-  STA $0300,X  --store that byte in location $300+X
9008-  INX          --increment the X register by 1
9009-  BNE $9002    --if X does not equal 0, branch back to $9002
900B-  JMP $030E    --jump to the relocated routine at $30E
```

● Section 2 uses the RWTS to read tracks 20-22 into $9000-$B6FF. We will be using the IOB (Input Output Block) that DOS normally uses. It is located at $B7E8. The IOB contains the information the RWTS needs as far as Track and Sector to read and where to put it in memory.

```
900E-  JSR $FE89    --these two instructions disconnect DOS so we can
9011-  JSR $FE93      load over the DOS area without any problem
9014-  JSR $FC58    --clear the text screen
9017-  LDA #$27     --this is used for a counter to specify how many
9019-  STA $B7FF      sectors we want to read from the disk
901C-  LDA #$00     --store a $00 byte in the IOB locations
901E-  STA $B7EB      for the volume number of the disk (0 matches
9021-  STA $B7F0      anything) and the low byte of the data buffer
9024-  LDA #$90     --this is the high byte of the data buffer
9026-  STA $B7F1      where we will be loading into ($9000)
9029-  LDA #$20     --this specifies track $20 where we will be
902B-  STA $B7EC      reading from
902E-  LDA #$0F     --we will begin reading from sector F and the
9030-  STA $B7ED      location to store the sector in is $B7ED
9033-  LDA #$B7     --the accumulator and Y-register must contain the
9035-  LDY #$E8      high and low order bytes of the IOB respectively
9037-  JSR $B7B5    --call the RWTS subroutine to read a sector
903A-  INC $B7F1    --increment the high order byte of the data buffer
903D-  DEC $B7FF    --decrement our sector counter at $B7FF
9040-  BEQ $904D    --if the counter is 0 branch to the next section
9042-  DEC $B7ED    --decrement the sector until it becomes negative
9045-  BPL $9033    --branch if plus (positive) to read next sector
9047-  INC $B7EC    --increment to next track
904A-  JMP $032E    --jump back and continue reading
```

...continued on page 26

3) At 96FA is a JMP $801. This is the exit into the next boot stage that loads into $800. We will change this to jump to $9801 and at $9801 we will put a routine to turn off the disk drive motor this is at $9801 is that we will be relocating boot 1 from $800 to $9800 later.

96FA:98 N 9801:AD E8 C0 4C 59 FF

4) We are ready to execute the boot and when the Apple beeps, boot 1 will be in at $800 and can be modified.

9600G

5) What we have to do now is move boot 1 at $800 up to $9800 to modify it. If we modify it at $800, it will just be overwritten next time we reboot. Boot 1 occupies only one page and is easy to relocate.

9800<800.8FFM

6) If you look at $988C you will see a JMP $BB29. This is the jump to the next and final boot stage of Wavy Navy. We must change this so it jumps to the monitor instead.

988D:59 FF

7) Once again, we can execute the boot at $9600.

9600G

8) Boot 2 is now loaded in. Listing through memory, nothing strikes me until I get toward the end of page $BE. At $BEDC is a JMP $800 and this is the jump to the start of the game (I'm surprised Sirius made this Boot Code so easy to trace; it's not like them). All we have to do is change the JMP $800 to a JMP $FF59 and boot 2 will jump to the monitor instead of jumping to the start of the game.

BEDD:59 FF

9) Boot 1 at $9800 is still exiting into the monitor, so that must be changed back so it will jump to boot 2 at $BB29.

988D:29 BB

10) If we were to execute the boot right now the game would load and..... begin running. Why? At $982F is a LDA #$BB which tells boot 1 to load boot 2 at page $BB which is $BB00. Even though we modified boot 2, it will get loaded right over. All we have to do is change the $BB to a $DB and it will try to load boot 2 in at $DB00, which is ROM, and nothing will be loaded over.

9830:DB

11) Now everything is set. When the boot is executed, the game will load in and return control to us.

9600G

12) Once the game is loaded in, we can save the first section of the game from $800 to $8FFF. However, before we reboot we have to move page $8 to a safe place in memory so it will not be overwritten when we reboot DOS.

9000<800.8FFM

13) You should have 2 blank initialized disks with no HELLO program. I will refer to them as disk 1 and disk 2. Insert disk 1 and reboot DOS.

```
C600G
```

14) Now we have to enter the monitor again and move page $8 back from $9000.

```
CALL -151
800<9000.90FFM
```

15) Before we can save the first part of Wavy Navy, we have to patch DOS so we can save a file longer than $7FFF bytes, then save the file.

```
A964:FF
BSAVE WAVY1,A$800,L$8800
```

16) What we have to do next is get the second part of the game in memory. To do this we'll have to repeat steps 2 through 11 (come on now, it's not that bad) to Boot Code Trace the disk again.

17) When you finish step 11 and the game is loaded in, move $9000 through $BAFF down to safe memory like $2000 so it can be saved.

```
2000<9000.BAFFM
```

18) Now insert disk 1 and reboot DOS.

```
C600G
```

19) Just to be safe, save section 2 of the game. That way if the memory gets accidentally overwritten, you won't have to do the beloved Boot Code Trace again.

```
BSAVE WAVY2,A$2000,L$2B00
```

20) Now enter Hexdump 1 at $9000. This is the routine for reading the second part of the game directly from the disk. See the sidebar for an explanation and listing to check your typing with.

Hexdump 1

```
9000: A2 00 BD 00 90 9D 00 03
9008: E8 D0 F7 4C 0E 03 20 89
9010: FE 20 93 FE 20 58 FC A9
9018: 27 8D FF B7 A9 00 8D EB
9020: B7 8D F0 B7 A9 90 8D F1
9028: B7 A9 20 8D EC B7 A9 0F
9030: 8D ED B7 A9 B7 A0 E8 20
9038: B5 B7 EE F1 B7 CE FF B7
9040: F0 0B CE ED B7 10 EC EE
9048: EC B7 4C 2E 03 A9 04 8D

9050: F1 B7 8D FF B7 CE ED B7
9058: A9 B7 A0 E8 20 B5 B7 EE
9060: F1 B7 CE ED B7 CE FF B7
9068: D0 EE A2 00 BD 00 04 9D
9070: 00 B7 BD 00 05 9D 00 B8
9078: BD 00 06 9D 00 B9 BD 00
9080: 07 9D 00 BA E8 D0 E5 4C
9088: 00 08
```

21) When you finish entering it save it to disk.

```
BSAVE HEXDUMP1,A$9000,L$8A
```

22) Now we'll modify this routine to write (instead of reading) the second section of the program to tracks $20 through $22.

```
9018:2B N 9025:20 N 904D:60 N B7F4:02
```

23) Now insert disk 2 and execute the routine.

```
9000G
```

24) Once this is done, run your sector editor and read Track $11 Sector $0 of disk 2. At byte $B8 you should see the following:

```
FF FF 00 00 FF FF 00 00 FF FF
```

Enter 10 consecutive $00 bytes here to cover up the $FF bytes and then rewrite the sector. This blocks out tracks $20 through $22 so they will not be used by any other file written to disk 2 (see DOS manual, page 133).

25) We are now entering the home stretch and the final step is transferring the main program to disk 2 along with the loader program. First we want to insert disk 1 and load section 1 of the game and Hexdump 1.

```
BLOAD WAVY1
BLOAD HEXDUMP1
```

26) Now all we have to do is put a jump to our $9000 routine at $7FD, patch DOS to save a 139 sector file, and save the program to disk 2 (Remember to insert disk 2). Disk 1 will no longer be needed.

```
7FD:4C 00 90
A964:FF
BSAVE WAVY NAVY,A$7FD,L$8890
```

If you wish to make a back-up of your unprotected version, your best bet is to copy the whole disk. Otherwise, it would involve copying the file, copying tracks $20 through $22, and modifying the track bit map for those tracks again.

A neat trick: Partway into the first level, press Reset a few times very quickly. This should leave you in Applesoft. Then type **CALL 2053** to run the game again. Your boat should be out of sync with the waves and you can hide underwater to avoid being hit. Also if you wish to play a game with unlimited ships, make the following modifications:

```
BLOAD WAVY NAVY
6A06:00 N 6693:00 N 6B10:00
1E64:EA EA N 1E97:EA EA
```

Then type **7FDG** to start the game. Have fun but don't get seasick!!

The Loader (continued)

● Section 3 is basically the same as Section 2. The RWTS is used to read the last 4 pages of part 2 of Wavy Navy into the text page buffer at $400 because we ran out of memory at $B700.

```
904D-  LDA #$04     --specify to read into page 4 which is the text
904F-  STA $B7F1      page buffer
9052-  STA $B7FF    --the 4 also specifies how many sectors to read
9055-  DEC $B7ED    --decrement the sector counter to the next sector
9058-  LDA #$B7     --as explained earlier the accumulator and
905A-  LDY #$E8       Y register must contain the address of the IOB
905C-  JSR $B7B5    --once again, execute the RWTS routine
905F-  INC $B7F1    --increment the page number of the data buffer
9062-  DEC $B7ED    --decrement the sector number
9065-  DEC $B7FF    --decrement the sector counter until it reaches 0
9068-  BNE $9058    --if not yet 0, continue to read
```

Now that the RWTS is no longer needed, we can move the memory that occupies the text page (at $400) to $B700 where it belongs and finally, jump to the start of the game at $800.

```
906A-  LDA #$00     --load the X register with 0
906C-  LDA $0400,X  --load a byte from memory location $0400+X and
906F-  STA $B700,X    store it in location $B700+X
9072-  LDA $0500,X  --this works the same way as the last two
9075-  STA $B800,X    instructions except it moves page $05 to $B8
9078-  LDA $0600,X
907B-  STA $B900,X
907E-  LDA $0700,X
9081-  STA $BA00,X
9084-  INX          --increment the X register
9085-  BNE $906C    --if X does not equal 0 branch back to $906C
9087-  JMP $0800    --memory move complete.  Jump to start of game
```

Kindercomp

by B. Croome

Spinnaker Software Corp.
215 First Street
Cambridge, MA 02142

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

Kindercomp is a "collection of six learning games" only two of which, "MATCH" and "SEQUENCE", have any real merit as they are interactive learning...a third is a hard (for an child) drawing program and the others are just pretty graphics.

I was asked by a friend of mine to attempt to deprotect this disk as he could not make back-ups of it for his children to use. Despite the fact that Spinnaker cautions the user...a child, yet, to "not bend the disk or touch the exposed part of the disk" and despite the fact that this program was actually designed for children, the disk was protected...downright silly!

On to the deprotection. On booting the disk the Applesoft prompt "]" was seen so I felt we should have a near normal DOS in action. The boot seemed normal with no funny skips or half-tracks so I assumed it used normal tracks. I booted it again and this time when I got the title page I hit Control-Reset and was immediately dropped into an Applesoft prompt.

I attemped to CATALOG the disk and was successful in receiving a full catalog with Applesoft and Binary files. Those of you familiar with Beagle Bros' Apple Mechanic will recognize their files along with several others. I then attempted to LIST the program in memory...it was still there and intact. I attempted to LOAD another Applesoft file and that worked fine too. I then attempted to SAVE the file in memory...nothing! This confirmed that the DOS and RWTS, while similar enough to Applesoft to run a standard Beagle Boys program, had also been modified somewhat. The program MENU will act just fine as the Hello program.

Using the Nibble Editor from Copy II Plus, I examined the various tracks and sectors on the disk and determined that tracks 0 to 2 had had the epilog byte changed from normal and tracks 3 on had the Address prologs changed from D5 AA 96 to B7 AA 96 (track 7 was just a bit weird).

Since everything seemed normal except the DOS, Super IOB was the perfect candidate for the job. The controller below will copy tracks $3-$22, skipping track 7. Poof... a deprotected disk!

Step by Step

1) Format a blank disk with a normal DOS. Use "MENU" as the "hello" filename.

INIT MENU

2) Use the Super IOB controller (installed in Super IOB 1.5) below to copy from the original Kindercomp disk to the blank disk.

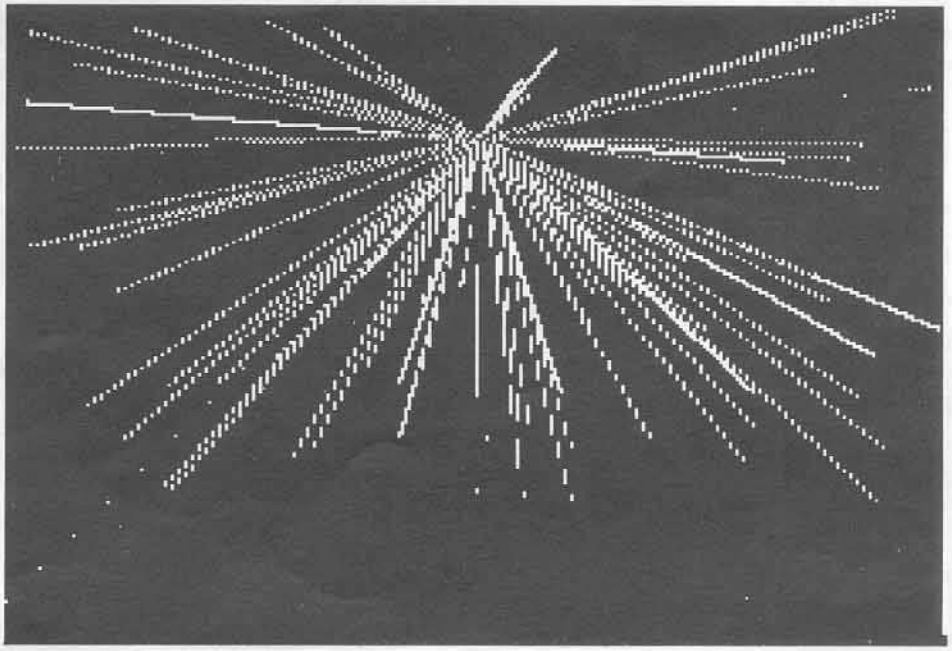
3) You're done! Hide the original disk.

controller

```
1000 REM KINDERCOMP
1010 TK = 3 :LT = 7 :ST = 15 :LS = 15 :CD = WR
     :FAST = 1
1020 RESTORE : GOSUB 190 : GOSUB 490 : GOSUB
     610
1030 GOSUB 230 : GOSUB 490 : GOSUB 610 : IF LT
     = 7 THEN LT = 35
1035 IF PEEK (TRK ) = LT THEN 1050
1040 TK = PEEK (TRK ) + (TK = 3 ) :ST = PEEK
     (SCT ) : GOTO 1020
1050 HOME : PRINT "COPYDONE" : END
5000 DATA 183 ,170 ,150
```

controller checksums

```
1000  - $356B     1035  - $74FB
1010  - $D424     1040  - $4DF0
1020  - $9D0A     1050  - $B14B
1030  - $268F     5000  - $7678
```



Flight Simulator

by Eric Sunshine

It is likely that changes have been made to the code residing at $A7E0-$B2DF in some versions of Flight Simulator][by SubLOGIC. As such, step 7 of my softkey (COMPUTIST No. 36) must be modified so as to capture the "language card data" from these altered versions. Although I am unable to supply the "exact" modifications, I can present helpful information, and "land marks", of sorts, which may aid in overcoming this obstacle.

Begin by examining the code at $1D00 after completing step 6 of the softkey.

```
1D00- STX $1E08 Store SLOT * 16
1D03- LDA #$00
1D05- STA $1E00
1D08- JSR $1EC4 Load DOS
1D0B- BCS $1D30 Error, go to error handler
1D0D- LDA #$E0
1D0F- STA $1E03 Store $A7E0 as destination
1D12- LDA #$A7  address for upcoming read
1D14- STA $1E04
1D17- LDA #$22  Start at Quarter $22
1D19- STA $1E01
1D1C- LDA #$00
1D1E- STA $1E02
1D21- JSR $1EAD Read
1D24- BCS $1D30 Error, go to error handler
1D26- LDA $1E01
1D29- CMP #$25  Done with read?
1D2B- BCC $1D21 No, continue reading
1D2D- JMP $A7E2 Yes, jump to "main boot"
```

Look for the jump to the "main boot". On the version for which the softkey was written, the code at $A7E2 disassembles to reveal a jump to location $AA4D.

```
A7E2-  4C 4D AA   JMP $AA4D
A7E5-  4C 4D AA   JMP $AA4D
```

The second, identical JMP, is part of the RESET handler. Next, disassemble the code to which this jump is made. The softkeyed version reveals a series of subroutine calls and error checks.

```
AA4D- NOP
AA4E- LDX #$3F
AA50- TXS
AA51- JSR $1EC4 Load DOS
AA54- BCS $AA70 Error, go to error handler
AA56- JSR $AAA5
AA59- BCS $AA70
AA5B- JSR $ABB5
AA5E- BCS $AA70
AA60- JSR $AC9B Check for, & load RAM Card
AA63- BCS $AA70 Error, go to error handler
AA65- JSR $A71C
AA68- BCS $AA70
AA6A- JSR $AD2B
AA6D- JMP $AA76
```

Disassemble the code listed for each JSR. Note the call which references the language card. On the softkeyed version, this was found in the subroutine at $AC9B.

```
AC9B- LDA $C083 Read/write enable RAM card
AC9E- LDA $C083
ACA1- LDX #$00
ACA3- STX $D000
ACA6- CPX $D000 Is RAM card present?
ACA9- BNE $AD2A No
ACAB- DEX
ACAC- BNE $ACA3
ACAE- LDA #$00  Yes. ACAEG in STEP 7
ACB0- STA $1E03   forces RAM card load
ACB3- LDA #$D0    STEP 7 changes this to a
ACB5- STA $1E04   $40 so it loads at $4000
ACB8- LDA #$00  Start with track 1
ACBA- STA $1E01
ACBD- JSR $1EB3 Read "language card data"
ACC0- BCS $ACCF
ACC2- LDA $1E01
ACC5- CMP #$22  Done reading?
ACC7- BCC $ACBD No, read more
ACC9- LDA #$01  Yes, indicate RAM card
ACCB- STA $08B2 presence
ACCE- CLC
ACCF- RTS
```

Notice the two locations referenced by step 7 of the softkey. If in fact, the code has been moved and/or altered, the memory locations will change. The $40 must replace the $D0 in the subroutine. Likewise, the routine must be activated from the correct spot so as to force load the RAM card. Replace the ACAEG with the appropriate address.

Step 16 of the softkey may also need alteration. Search the disk for a sequence of bytes which refer to the language card. The bytes AD 83 C0 AD 83 C0 A2 00 should be found only once. On the version from which the softkey was written, these bytes were found on Track $09, Sector $0C. The code which checks for and loads the RAM card, must be modified so that it calls the RAM card load routine only once. This was accomplished on the softkeyed version by replacing bytes $E2 through $E8 of Track $09, Sector $0C with the NOP instruction.

```
AC9B-  AD 83 C0   LDA $C083
AC9E-  AD 83 C0   LDA $C083
ACA1-  A2 00      LDX #$00
ACA3-  8E 00 D0   STA $D000
ACA6-  EC 00 D0   CPX $D000
ACA9-  D0 7F      BNE $AD2A
ACAB-  CA         DEX
ACAC-  D0 F5      BNE $ACA3
ACAE-  A9 00      LDA #$00
ACB0-  8D 03 1E   STA $1E03
ACB3-  A9 D0      LDA #$D0
ACB5-  8D 04 1E   STA $1E04
ACB8-  A9 00      LDA #$00
ACBA-  8D 01 1E   STA $1E01
ACBD-  20 B3 1E   JSR $1EB3
ACC0-  B0 0D      BCS $ACCF
ACC2-  EA         NOP
ACC3-  EA         NOP
ACC4-  EA         NOP
ACC5-  EA         NOP
ACC6-  EA         NOP
ACC7-  EA         NOP
ACC8-  EA         NOP
ACC9-  A9 01      LDA #$01
ACCB-  8D B2 08   STA $08B2
ACCE-  18         CLC
ACCF-  60         RTS
```

Replace the appropriate bytes with $EA in the sector in which the code is located.

The Product Monitor

Credits/Intro

by Jeff Hurlburt

A department is born. In recognition of the furious product development activity which has swept the Apple world of late, COMPUTIST has opted for more frequent, fresher perusals of just what's in store for you and your computer. So, sit back and and enjoy.

Ratings

★★★★★ Superb
★★★★ Excellent
★★★ Very Good
★★ Good
★ Fair
Poor
Bad
Defective

Apple //gs Technical Reference (Book)

By: Michael Fischer
Price: $19.95
Publisher: McGraw-Hill
Rating: ★★★

A few months ago I had the pleasure of reviewing Fischer's book on programming the 65816 and so was very interested in his latest //gs effort. "Technical Reference", to begin with, is somewhat misleading. True, there is plenty of specific technical information, but also some important gaps which should be noted at the outset. Except for slot and connector pin-outs you will find nothing specific relating to hardware (no chip data, schematics, or timing diagrams). Though some attempt is made at a system overview, the section is too sketchy to be of much value; and, finally, if you want an in-depth exposition of the //gs monitor (e.g. an annotated listing), you will have to look elsewhere.

Essentially a system firmware reference, the McGraw-Hill text leads off with good coverage of memory allocation and soft switches, lists monitor entry points (addresses and routine names), and provides some assembly language examples for managing Battery RAM and modifying Control Panel settings. Very little is said about the mini-assembler; but monitor commands are listed and described.

Unquestionably, the book's greatest strength shows in discussions of text and graphics displays, and the large sections dealing with Quickdraw and the Desktop Bus (windows, dialogue boxes, "controls", ...). Probably, the coverage given either of the latter two areas is worth the book's price. Other topics receiving decent treatment include the Sound Manager, ProDOS 16, disk I/O, the Event Manager, Desk Accessories, interrupts, and the "Standard Apple Numerics Environment'.

In case some of the above sounds like Greek, be assured that you are not alone. Fortunately, you have the author's conversational style, program examples, and copious charts, tables, and other aids. All go a long way towards making things clear. The //gs, chiefly due to its considerable resident firmware, is not a simple machine. Fischer's *Apple //gs Technical Reference* is both an excellent way to get started and a valuable long-term resource.

SpinDizzy

Publisher: Electric Dreams, from Activision
Requires: 64K Apple][series, joystick
Price: $34.95
Rating:

Among the more difficult of hi-res animation tasks is the real-time movement of some familiar object (like a marble or top) along a simulated 3-D surface. Everyone knows how such an object is supposed to both look and sound; any slip-ups totally ruin the effect.

Undaunted by such concerns, the Electric Dreams group has produced *Spindizzy*, a 3-D maze-running contest. Ostensibly, you and/or another player are engaged to map a strange new world repleat with ramps, lifts, trampolines, water hazards, and gaping chasms. Your mapping device is a stick-controlled "robot" shaped like a gyroscope (optionally, a marble or top; shape has no effect on play).

The race is against a countdown timer, not each other, so there is a single score expressed in % of the maze mapped and % of available "energy-boosting" (time adding) jewels collected. Falling into a hazard subtracts from player time and, in a two-player game, transfers control. Selecting the single-player, slow speed variation, a beginner can expect five or ten minutes of action before the countdown reaches zero. Restart, happily, occurs without delay.

While the option to view a rough, low-res map of the maze is helpful, the fact that screens are switched instead of scrolled makes it difficult to plan exits, costing frequent, game-delaying drops off the surface. Even worse, maze sections are arranged so that cardinal points face screen corners. (i.e. stick UP moves your robot towards the upper right corner!) All of which might be forgiven if the controls were moderately responsive, instead of feeling like a bowl of jello.

A game beset by low production values, *Spindizzy* displays, at best, circa '82 graphics technique and virtually nothing in the way of sound. Add this one to your "avoid-if-I-can" list.

Updates Etc.

Tass Times in Tonetown (which was very favorably reviewed) is available in super hi-res for the //gs. The 3.5" mini-disk (256K required) version sells for $44.95. A great showpiece for the //gs, the game rates a solid ★★★★

In the COMPUTIST No. 40 (favorable) review of *War Or Piece? You Decide!* there was an error in the Bright Ideas ordering address. The corrected information appears with this month's vendors.

Vendors

Activision
POB 7287
Mountainview, CA 94039
(800) 227-9759
In California: (415) 940-6044

Bright Ideas
52 Exeter Street
Portland, ME 04102
(207) 777-1330

McGraw-Hill
2600 Tenth Street
Berkeley, CA 94710
(415) 548-2805

The Book Of Softkeys

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

The Book of Softkeys

Volume I: $7.95
Volume II: $12.95
Volume III: $17.95

Because of the demand for the information contained in those rare issues of Hardcore COMPUTIST, we went deep into the archives to come up with a collection of softkeys originally contained in issues 1 through 15:

The Book of Softkeys

We compiled them into books which contain all the softkeys and some of the more choice articles from these out-of-print issues.

Each book is over 150 pages long, brimming with deprotection information and makes a economical alternative to those rare (and largely unavailable) back issues of Hardcore COMPUTIST.

Come, discover five issues of COMPUTIST history in each book.

Use the enclosed order form to get yours *today!*

Softkeys covered in each volume

Volume I: Issues 1-5

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

Volume II: Issues 6-10

contains softkeys for: Apple Cider Spider | Apple Logo | Arcade Machine | The Artist | Bank Street Writer | Cannonball Blitz | Canyon Climber | Caverns of Freitag | Crush, Crumble & Chomp | Data Factory 5.0 | DB Master | The Dic*tion*ary | Essential Data Duplicator I & III | Gold Rush | Krell Logo | Legacy of Llylgamyn | Mask Of The Sun | Minit Man | Mouskattack | Music Construction Set | Oil's Well | Pandora's Box | Robotron | Sammy Lightfoot | Screenwriter II v2.2 | Sensible Speller 4.0, 4.0c, 4.1c | the Spy Strikes Back | Time Zone v1.1 | Visible Computer: 6502 | Visidex | Visiterm | Zaxxon | Hayden Software | Sierra Online Software | **PLUS** the complete listing of the ultimate cracking program...Super IOB 1.5 | **and more!**

Volume III: Issues 11-15

contains softkeys for: Alien Addition | Alien Munchies | Alligator Mix | Computer Preparation SAT | Cut And Paste | Demolition Division | DLM (Developement Learning Materials) software | EA (Electronic Arts) software | Einstein Compiler version 5.3 | Escape From Rungistan | Financial Cookbook | Flip Out | Hi-Res Computer Golf II | Knoware | Laf Pak | Last Gladiator | Learning With Leeper | Lion's Share | Master Type v1.7 | MatheMagic | Minus Mission | Millionaire | Music Construction Set | One On One | PFS software | PS (Penguin) Software | The Quest | Rocky's Boots | Sabotage | Seadragon | Sensible Speller IV | Snooper Troops II | SoftPorn Adventure | Stickybear series | Suicide | TellStar | Tic Tac Show | Time Is Money | Transylvania | Type Attack | Ultima III Exodus | Zoom Graphics | Breaking Locksmith 5.0 Fast Copy | **PLUS** feature articles on | Csaver | The Core Disk Searcher | Modified ROMs.

To order: The Book of Softkeys

☐ **Volume I - $7.95 + $2 shipping/handling**
☐ **Volume II - $12.95 + $2 shipping/handling**
☐ **Volume III - $17.95 + $2 shipping/handling**

☐ **All three volumes!!**
$30.00 + $2 ship/handling

Name_____

Address_____

City _____ State ____ Zip _____

Country _____ Phone _____

VISA _____-_____-_____-_____ Exp._____

Signature_____ CP45

Foreign orders (except Canada and Mexico) please add $5 for shipping and handling. Washington residents add 7.8% sales tax.
Most orders are shipped within 5 working days, however, please allow 4-6 weeks delivery. **US Funds drawn on US banks only.**

Send to: Book of Softkeys PO Box 110846-T Tacoma, WA 98411
(206) 474-5750

the COMPUTIST shopper...

Books

Educational Systems
- Microref for Wordstar....$12.00
- Microref for Supercalc....$12.00
- Microref for Visicalc....$12.00

Software

Academy Software, Inc
- Typing Tutor + Word Invaders (//e)....$16.00

Accolade
- Dambusters....$24.50
- Hardball....$24.50
- Fight Night....$24.50
- Law of the West....$24.50

Addison-Wesley
- Smart Eyes....$48.00
- The Hobbit....$28.00
- The Fellowship of the Ring....$32.00

Activision
- Greeting Card Maker....$32.00
- Term Paper Writer....$48.00
- Writer's Choice....$39.75
- Filer's Choice....$39.75
- Planner's Choice....$39.75
- Hacker....$32.00
- Ghostbusters....$32.00
- Tass Times in Tonetown....$32.00
- Space Shuttle: Journey into Space....$32.00
- The Rocky Horror Show....$27.75
- Shanghai....$32.00
- Little People Discovery Kit....$32.00
- Gamemaker....$39.75
- Labyrinth....$32.00
- Borrowed Time....$32.00
- Portal....$36.00
- Alter Ego (male)....$39.75
- Alter Ego (female)....$39.75
- Mindshadow....$32.00
- Murder on the Mississippi....$32.00
- Hacker II....$32.00

Advanced Ideas
- The Game Show....$32.00
- Tic Tac Show....$32.00
- Wizard of Words....$32.00

American Education Computer
- Spanish....$16.00
- Spelling....$39.75
- Phonics....$39.75
- Learn to Read....$39.75
- Reading Comprehension....$39.75
- U.S. History....$16.00
- U.S. Geography....$16.00

Artsci
- **Magic Window //e....$106.00**
- **Magic Window II....$106.00**
- Magic Office System....$176.00
- Magic Window II extra (128K)....$141.50

Ashton-Tate
- DBase II (requires Z80)....$415.00

Avalon Hill
- Statis Pro Baseball....$28.00
- Mission on Thunderhead....$20.00
- Dreadnoughts....$24.00
- Under Fire....$47.50

Bantam Electronics Publishing
- Card & Party Shop....$32.00
- Comic Strip Maker....$32.00

Baudville
- Award Maker Plus....$28.50
- Take 1 Delux....$42.00
- Blazing Paddles....$24.25
- Prince....$35.25
- Ted Bear's Rainy Day Games....$21.50
- Video Vegas....$21.50

Beagle Bros
- Apple Mechanic....$21.00
- Beagle Basic Utility....$25.75
- Double Take....$25.75
- Utility City....$21.00
- DOS Boss Utility....$17.25
- Silicon Salad....$17.50
- DiskQuick....$21.00
- Triple Dump....$28.00
- Fat Cat....$24.75
- D Code....$28.50
- Power Print....$28.50
- Extra K....$28.50
- Pro-Byter....$24.75
- Big U....$24.75
- Macroworks....$24.75
- Super Macroworks....$35.25
- **Global Program Line Editor....$35.25**
- Beagle Graphics....$42.75
- Frame-UP....$21.00
- Shape Mechanic....$28.50
- Font Mechanic....$21.50
- Minipix Disk 1....$21.50
- Minipix Disk 2....$21.50
- Minipix Disk 3....$21.50
- Alpha Plot....$25.50
- Beagle Screens....$25.50
- Beagle Compiler....$53.25
- Beagle Bag....$21.00

BPI Systems
- Payroll....$198.00
- Accounts Receivable....$198.00
- Accounts Payable....$198.00
- General Accounting (ProDOS)....$198.00
- Payroll (ProDOS)....$198.00
- Accounts Receivable (ProDOS)....$198.00
- Accounts Payable (ProDOS)....$198.00
- Inventory Control (ProDOS)....$198.00

Broderbund Software
- Science Tool Kit Master Module....$55.50
- Science Tool Kit Module 1: Speed and Motion....$32.00
- Science Tool Kit Module 2: Earthquake Lab....$32.00
- Where in the USA is Carmen San Diego?....$36.00
- Where in the WORLD is Carmen San Diego?....$32.00
- Type....$36.00
- On Balance....$79.50
- Bank Street Writer 64K....$55.50
- Bank Street Mailer 64K....$55.50
- Bank Street Mailer 128K....$55.50
- Bank Street Filer 64K....$55.50
- Bank Street Filer 128K....$55.50
- Bank Street Speller....$55.50
- Dazzle Draw....$47.50
- The Print Shop....$39.75
- Print Shop Companion....$32.00
- Print Shop Graphics Library Disk 1....$20.00
- Print Shop Graphics Library Disk 2....$20.00
- Print Shop Graphics Library Disk 3....$20.00
- Animate....$55.50
- The Toy Shop....$47.50
- Airheart....$28.00
- Lode Runner....$28.00
- Championship Lode Runner....$28.00
- Breakers....$36.00
- Karateka....$28.00
- Captain Goodnight and the Islands of Fear....$28.00

DLM Inc.
- Number Farm....$24.00
- Create With Garfield....$24.00
- Create w/Garfield - Delux....$32.00
- Teddy Bear-rels of Fun....$32.00
- Fish Scales....$24.00
- Meteor Multiplication....$27.00
- Alligator Mix....$27.00
- Decimal Discovery....$36.75
- Writing Adventure....$37.00

EPYX
- Temple of Apshai Trilogy....$32.00
- Championship Wrestling....$32.00
- Destroyer....$32.00
- The Movie Monster Game....$32.00
- World Games....$32.00
- Summer Games....$32.00
- Summer Games II....$32.00
- Winter Games....$32.00
- World's Greatest Baseball Game....$32.00
- World's Greatest Football Game....$32.00

Infocom
- Zork I....$32.00
- Wishbringer....$32.00
- Suspended....$39.75
- Planetfall....$32.00
- Seastalker....$32.00
- A Mind Forever Voyaging....$36.00
- Enchanter....$32.00
- Sorcerer....$36.00
- Spellbreaker....$39.75
- Trinity....$32.00
- Fooblitzky....$32.00
- Ballyhoo....$32.00
- Hitchhiker's Guide....$32.00
- Moonmist....$32.00
- Starcross....$39.75

Micropro International
- Wordstar Professional....$349.50
- Wordstar....$247.00

Microprose Software
- Decision in the Desert....$32.00
- Crusade in Europe....$32.00
- Conflict in Vietnam....$32.00
- Gunship....$32.00
- F-15 Strike Eagle....$28.00
- Solo Flight....$28.00
- Silent Service....$28.00

Hardware

AAA International
- 5 ¼ inch Wooden Diskcase (Holds 70 5 ¼ inch disks)....$20.25
- 3 ½ inch Wooden Diskcase (Holds 30 3 ½ inch disks)....$16.00

Abati
- Letter-Quality Parallel Printer (LQ-20P) 132 COL, 18 CPS, with tractor....$298.50

Acrylic Arts, Inc.
- Universal printer stand....$17.00

Amdek
- Amdisk I - 3" disk drive....$274.50
- Video 300A Amber monitor....$183.00
- Tilt/Swivel for color 300-730....$30.00

AST Research-Apple Division
- Multi I/O (with clock, calendar, 2 serial ports)....$157.50
- Multi I/O (with clock & calendar)....$130.50

C H Products
- Mach III w/Fire Button (//e)....$44.00
- Mach II Joystick (//e)....$36.50
- Mach III w/Fire Button (][,][+)....$44.00
- Mach II Joystick (][,][+)....$36.50

Compu Cable
- Apple][6 ft parallel cable....$30.00

Hayes Microcomputer Products
- Smartmodem 300 w/Smartcom I (//c)....$222.00
- Micromodem //e w/Smartcom I....$186.00

Kraft Systems
- Apple Joystick (3 buttons)....$42.75

Panasonic
- Dot Matrix parallel printer (KX-P1081) 80 COL, 120 CPS....$298.50
- Dot Matrix parallel printer (KX-P1091) 80 COL, 160 CPS....$389.25
- Dot Matrix parallel printer (KX-P1592) 132 COL, 180/38 CPS....$627.00
- Dot Matrix parallel/serial printer (KX-P1595) 132 COL, 240/51 CPS, 15K buffer$725.25

Video-7
- Color Enhancer //e w/Dazzle Draw....$162.00
- Color Enhancer //c w/Dazzle Draw....$117.00

How To Order

- **Circle your selection. If total order is less than $200, please add $2.00/item shipping & handling. (Most orders shipped UPS so please use street address.)**
- **Foreign orders please inquire as to appropriate shipping fees.**
- **Washington State residents please add 7.8% sales tax.**
- **Domestic orders over $200 FREE shipping.**
- **Offer good while supplies last. All products are for the Apple][unless otherwise specified**
- **Send Orders To:**

SoftKey Publishing
PO Box 110816-T
Tacoma, WA 98411
(206) 474-5750

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA _____-_____-_____-_____ Exp._____

Signature _____ CP45

Are you missing a piece of the picture ???!!!

Issue	Mag $4.75	Disk $9.95	Both $12.95
44	☐	☐	☐
43	☐	☐	☐
42	☐	☐	☐
41	☐	☐	☐
40	☐	☐	☐
39	☐	☐	☐
38	☐	☐	☐
37	☐	☐	☐
36	☐	☐	☐
35	☐	☐	☐
34	☐	☐	☐
33	☐	☐	☐
32	☐	☐	☐
31	☐	☐	☐
30	☐	☐	☐
29	☐	☐	☐
28	☐	☐	☐
27	☐	☐	☐
26	☐	☐	☐
25	☐	☐	☐
24	☐	☐	☐
22	☐	☐	☐
20	☐	☐	☐
19	☐	☐	☐
16	☐	☐	☐
8	•	☐	•
☆ 7	☐	☐	☐
Core 2	☐	☐	☐
1	☐	☐	☐
Core 1	☐	☐	☐
Core 3	☐	☐	☐
Computing 3	☐	•	•
Best of Hardcore Computing	•	☐	•
Core Special $10.00 (All three CORE magazines)	☐		

Some disks apply to more than one issue and are shown as taller boxes. Special "Both" disk & magazine combination orders apply to one issue and its corresponding disk.

Book Of Softkey titles are inserted for reader information only. Each book contains all of the softkeys appearing in the issues below the title line. Please consult current Book of Softkey ad for current prices and ordering information.

☆ We have a limited supply of these issues.
• Back issue is no longer available

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

Back Issues and Library Disk Rates

- US, Canada and Mexico back issue rate - **$4.75** each.
- All other Foreign back issue rate - **$8.75** each.
- US, Canada, Mexico library disk rate - **$9.95** each.
- All other Foreign library disk rate - **$11.94** each.
- "Both" disk and magazine rates for:
  US, Canada & Mexico - **$12.95** each combination.
  All other Foreign - **$18.95** each combination.

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

What is a library disk?

A library disk is a diskette that contains programs that would normally have to be entered by the user. Documentation for each library disk can be found in the corresponding issue.

• Library disks are available for all issues of COMPUTIST 1 thru 44. A description of the softkeys and programs covered in each issue is available upon request. Please send your name and addres along with a first class postage stamp (US $ .22).

Complete Your Collection!

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

CORE 2 Utilites: Dynamic Menu | High Res: Scroll Demo | GOTO Label: Replace | Line Find | Quick Copy: Copy |

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

Hardcore Computing 3 HyperDOS Creator | Menu Hello | Zyphyr Wars | Vector Graphics | Review of Bit Copiers | Boot Code Tracing | Softkey IOB | Interview with 'Mike' Markkula |

☐ **Please send the back issues indicated:**

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA MC _____-_____-_____-_____ Exp. _____

Signature _____ CP45

US funds drawn on US banks

☐ **Please send the following library disks:**

_____ _____ _____ _____

Send check or money order to:

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411.
(206) 474-5750

Most orders are shipped within 5 working days, however please allow up to 4 weeks delivery for some orders. Most orders shipped UPS, so please use street address. Offer good while supply lasts. In Washington state, add 7.8% sales tax.

Back Issues

US/Canada/Mexico - $4.75 ea Foreign - $8.75 ea

44 *Softkeys* | Arcade Boot Camp | Goonies | Zorro | Coveted Mirror | Crimson Crown | Compubridge | Fleet System 3 | Microwave | Escape | Catalyst 3.0 | Number Farm | Alphabet Circus | Joe Theisman's Pro Football | Black Cauldron | International Gran Prix | *Features* | Making DOSless Utilities | Pixit Printer Drivers | *Core* | Review of Z-RAM Memory Expansion Board | Reading The Joystick | .....

43 *Softkeys* | Graphics Expander | Information Master | Certificate Maker | Elite | Catalyst 2.0 and 3.0 | Murder On The Mississippi | Temple Of Apshai Trilogy | Troll Associates programs | Spell It Regatta | Cdex Training programs | Think Fast *Features* | How to Write-Protect your Slot Zero | Capturing Locksmith 6.0 Fast Copy | Revisiting DOS to ProDOS and Back | *Core* | Computer Eyes / 2: a Review | *APTs* | Sword of Kadash & Rescue Raiders | Ultimaker IV | .....

42 *Softkeys* | Light Simulator | *Readers' Softkeys* | Beach-Head | Monty Plays Scrabble | Racter | Winnie the Pooh | Infocom Stuff, Kabul Spy, Prisoner II | Wizardry 1 & 2 | Lucifer's Realm | The PFS Series | Dollars and Sense | Strip Poker | Coveted Mirror | Wizard's Crown | The Swordthrust Series | Axis Assassin | Manuscript Manager | The Crown of Arthain | Address Book | Decimals 3.0 | Dragonfire *Features* | Auto Duel Editor | Wizard's Crown Editor | Questron Mapper | *Core* | The Games of 1986 in Review | *Adventure Tips* | Ultima IV | .....

41 *Softkeys* | The Periodic Table | Gemstone Warrior | Inferno | Frogger | *Readers' Softkeys* | Story Maker | Adventure Writer | Mummy's Curse | Zaxxon | The Quest | Pitfall II | H.E.R.O. | *Features* | A Two-Drive Patch for Winter Games | Customizing the Speed of a Duodisk | Roll the Presses Part Two: Printshop Printer Drivers | The Games of 1986 | ....

40 *Softkeys* | Adventure Writer | Mychess II | Raster Blaster | *Readers' Softkeys* | Cranston Manor | Ghostbusters | Designer's Pencil | E-Z Learner | The American Challenge | Crime Wave | Encyclopedia Britannica Programs | *Features* | Taking the Wiz out of Wizardry | Adding a Printer Card Driver to Newsroom | *Core* | The Games of 1986 | .....

39 *Softkeys* | MIDI/8 Plus | Homeword v2.1 | Borrowed Time | Amazon | Speed Reader][| *Readers' Softkeys* | Discovery! | M-ss-ng L-nks series | Donald Ducks's Playground | Mastering the SAT | Copy][Plus 4.4C | Master of the Lamps | One on One | Bridge Baron | A.E. | Great American Cross-Country Road Race | Computer Preparation for the SAT | Castle Wolfenstein | Luscher Profile | Skyfox | Silent Service | Echo Plus | Swashbuckler | Randamn | *Features* | Electronic Disk Drive Swapper | Abusing the Epilogues | Print Shop Companion's Driver Game | *Core* | Keyboard Repair | Fixing the Applesoft Sample Disk | *Hints* | Carmen Sandiego | .....

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 |

COMPUTIST PO Box 110846-T Tacoma, WA 98411

Looking for the Best Deal in Town?

How about ALL of our Super IOB controllers, (through 1986) in ONE package!

COMPUTIST developed the ultimate copy program to remove copy protection from software:

The Super IOB program.

Since the introduction of Super IOB, COMPUTIST has used this flexible program to deprotect (or partially deprotect) dozens of commercial programs with far ranging protection schemes.

Super IOB deprotects disks by using a modified RWTS (the subroutine in DOS which is responsible for the reading and writing of disk sectors) for reading from the protected disk and then using a normal RWTS for writing to the deprotected disk.

This package contains:

► TWO DISKS (supplied in DOS 3.3). Each disk contains at least 60 Super IOB Controllers including the standard, swap, newswap and fast controllers. Also included is **version 1.5 of Super IOB**, the Csaver program from COMPUTIST No. 13, and a Menu Hello Program that lists the available controllers and, when you select one, automatically installs it in Super IOB and RUNs the resulting program.*

► A reprint of **Disk Inspection and the Use of Super IOB**, from COMPUTIST No. 17. This article explains how to write your own Super IOB controllers.

► **COMPUTIST No. 32**, which contains an extensive article detailing the hows and whys of Super IOB v1.5 and at least 5 articles using the new Super IOB program.

- Several of the controllers deprotect the software completely with no further steps. This means that some programs are only minutes away from deprotection (with virtually no typing).
- The issue of COMPUTIST in which each controller appeared is indicated in case further steps are required to deprotect a particular program.**

The SUPER IOB Collection

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

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

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

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

☐ **Yes, please send me The Super IOB Collection**

Includes both disks with Super IOB version 1.5, COMPUTIST#32, PLUS a reprint of "Disk Inspection and the Use of Super IOB".

☐ **US/Canada/Mexico for $16 00**
☐ **Other Foreign for $20.00**

Send to: **Super IOB Collection**
PO Box 110846-T
Tacoma, WA 98411
(206) 474-5750

Name _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

VISA _____-_____-_____-_____ Exp._____

Signature _____ CP45

Most orders are shipped within 5 working days, however, please allow 4 to 6 weeks for delivery. Washington residents, please add 7.8% sales tax.

US funds drawn on US banks

*Requires at least 64K of memory.
**Although some controllers will completely deprotect the program they were designed for, some will not, and therefore require their corresponding issue of COMPUTIST to complete the deprotection procedure.