For The Serious User Of Personal Computers

COMPUTIST

Issue 57 July 1988

USA $3.75 Canada & Mexico $7.00 All Others $8.75

RDEX Features:

- **Electronic Art's Protection Language**—By Phil Goetz
- **How to find hidden code with EOR DiskScan**—By Phil Goetz
- **Lower Case & Infocom Games Revisited**—By Greg Poulos
- **Monsters of Might & Magic**—By Les Minaker
- **A Character Editor for Rings of Zilfin**—By Aaron Schoeffler
- **A Single Data Disk for all your Print Shop Graphics**—By Klaus Iden
- **The Product Monitor**—By Jeff Hurlburt

RDEX Softkeys:

American Challenge Arctic Fox Bard's Tale II BoulderDash Construction Set California Games Championship Wrestling Chief of Detectives/Drawing Conclusions Deep Space Dome Simplified Bookkeeping System Dr. Ruth's Computer Game of Good Sex Earth Orbit Stations Factory Galaxy Search/Predicting Outcomes Game Maker Hacker II Hardball IIgs Ikari Warriors Labyrinth Marble Madness Master Diagnostics IIe Math Blaster Mickey's Space Adventure Micro-Computer Learning Games Microzine #14 Microzine #24 Milliken Math Series Mind Prober M – ss – ng L – nks Morning Star Spelling Mountain Climbing/Cause and Effect Movie Monster Game Pond Race Track/Reading for Detail Reading Comprehension Main Idea & Details Rings of Zilfin Roadwar 2000 School Days/Inference Ski Crazed Softswitch Sub Mission Time Capsule/Reading Skills Tuesday Morning Quarterback Typewriter Where in Europe is Carmen Sandiego Wortgefecht Xevious

IBM RDEX:

Softkey: ■**Symphony v1.00** ■**TK!**

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

You have a LEGAL RIGHT to an unlocked backup copy of your commercial software

Our editorial policy is that we do NOT condone software piracy, but we do believe that users are entitled to backup commercial disks they have purchased.

In addition to the security of a backup disk, the removal of copy-protection gives the user the option of modifying programs to meet his or her needs.

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

> **..."It is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or *adaptation* of that computer program provided:**
>
> **1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or**
>
> **2) that such new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.**
>
> **Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner."**

United States Code title 17, §117

COMPUTIST

Publisher-Editor: Charles R. Haight
Graphics Editor: Ben R. Haight
Circulation: Karen Fitzpatrick
Advertising, call: (206) 474-5750
Publisher: SoftKey Publishing
Printing: Valco Graphics & Printing

● Address all advertising inquiries to: COMPUTIST; Advertising Department; PO Box 110816; Tacoma, WA 98411

● Mail all RDEX letters to:
COMPUTIST
Apple-RDEX or IBM-RDEX
PO Box 110846-K
Tacoma, WA 98411

● COMPUTIST does NOT purchase editorial material. The entire editorial content consists of information submitted to COMPUTIST for publication in the shared interests of all COMPUTISTs.

● Unsolicited material (manuscripts, letters to the editor, softkeys, A.P.T.s, playing tips, questions, etc., are assumed to be submitted as letters-to-the-RDEX-editor for publication with all and exclusive rights belonging to COMPUTIST.

● The volunteer and paid editorial staff assume no liability or responsibility for the products advertised in the magazine. Any opinions expressed by the authors are not necessarily those of COMPUTIST magazine, its staff or SoftKey Publishing.

● Apple® is a trademark of Apple Computers. IBM® is the IBM trademark.

SUBSCRIPTIONS: Rates (for 12 issues):

U.S.—$32	U.S. 1st Class—$45
Canada/Mex.—$45	Other Foreign—$75

● Send subscription inquiries to: COMPUTIST; Subscription Department; PO Box 110846-T; Tacoma, WA 98411

● For 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.

Attention Apple-users:

Why type those l-o-n-g listings when you can just Upgrade your present sub. to a Mag/5¼"Disk Combo subscription

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

Time to renew?

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

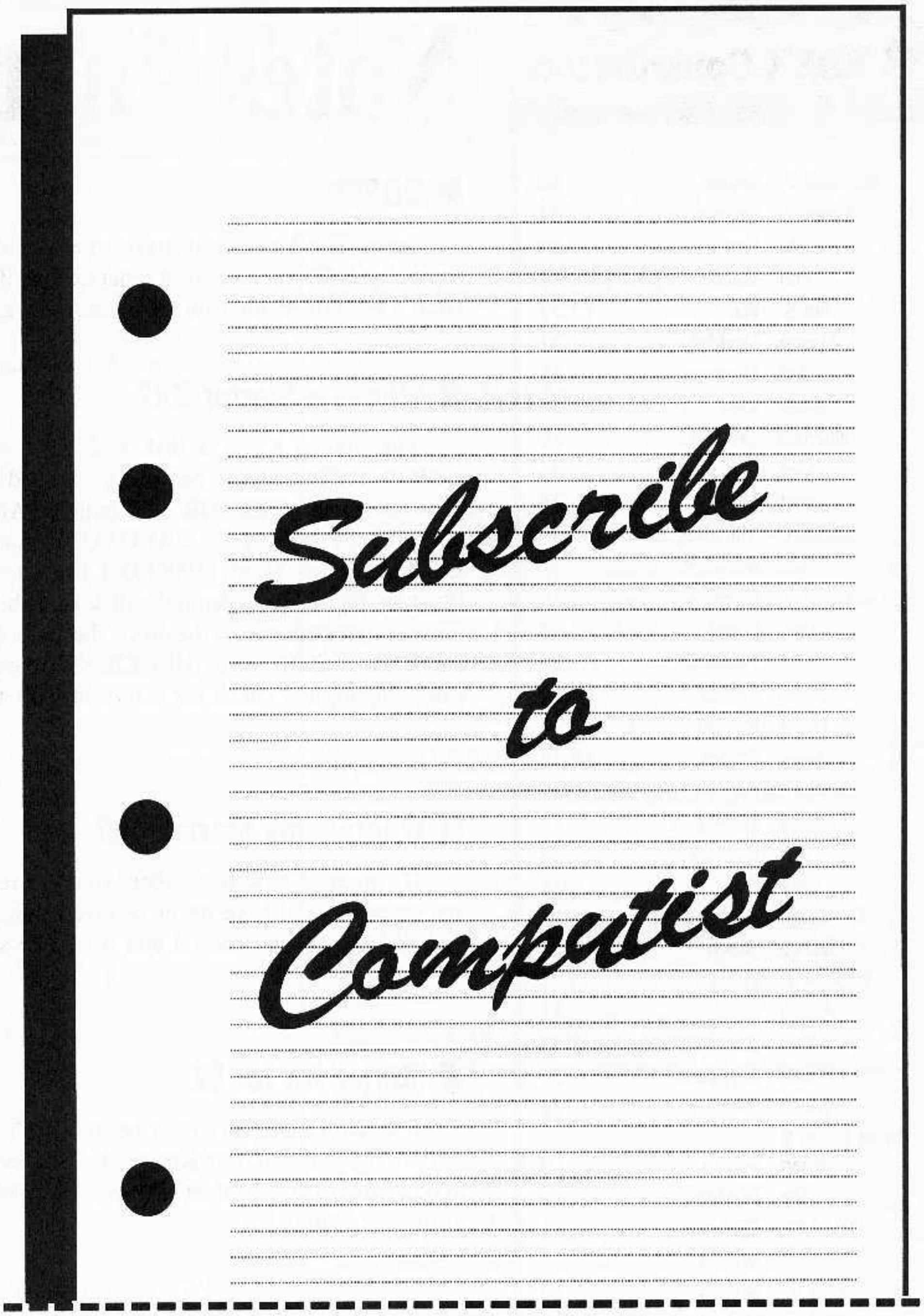
□ Use the order form on this page to renew, and don't forget to send your most current mailing label to help expedite your renewal

Moving soon?

■ 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 last mail label)

I have..□ an **Apple** IIe, IIgs or IIc computer
□ an **IBM** computer

Apple—IBM subscription...
□ U.S. - $32
□ U.S./Canada/Mexico First Class - $45
□ All other Foreign - $75

Apple mag-disk combination subscription...
□ U.S./Canada/Mexico First Class plus Library Disk - $100
□ All other Foreign plus Library Disk - $140

Name ______________________________ ID# __________

Address ______________________________

City ______________ State ______ Zip __________

Country ______________ Phone ______________

VISA ______-______-______-______ Exp. __________

Signature ______________________________ CP57

U.S. Funds drawn on U.S. bank. Allow 4—8 weeks for first issue. Mail to: COMPUTIST PO Box 110846-T Tacoma, WA 98411 (206) 474-5750

RDEX Contributors

Marshall P.	Brown	36
James E.	Bulman	21
A.	Evans	20
Phil	Goetz	10
Jim S.	Hart	17,22
Steven	Heckler	21
Jeff	Hurlburt	35
Klaus	Iden	33,37
Michael	Javorka	36
Alex	Lee	21
Scott	Lloyd	26
Michael A.	Mahaffey	25
Les	Minaker	30
Jose A.	Montano	14
Jack	Nissel	28
	Paladin	36
Keith	Parker	15
Doug	Parrish	15
Greg	Poulos	16,28
Carl D.	Purdy	26
Thomas	Purifoy	25
Jan	Recourt	24
Jolly	Roger	14
Dr. Leigh	Rowan-Kelly	24
Dr. George	Sabeh	27
Stephen J.	Scalia	27
Aaron	Schoeffler	31
Alan	Sheppard	24
David	Sheppard	25
Lewis J.	Shireman	36
David Burns & Brian	Sparks	16
Joe	Walters	9
David M.	Widman	21
Bill	Wilson	7
Paul R.	Wilson	20
Robert	Wilson	17
Name	Withheld	27
John	Wulfken	15
	Unk	22
	Unk	28
Leh-Wen	Yau	28
	Zorro	17

IBM RDEX Contributors

Mike Basford	44

Notes 'n things......

OOPS!

Seems that I made a mistake when I told you about the Starter Kit. The cost of the starter kit is $2 when ordered separately or if you have already received your free copy. Here's the information on the Starter Kit again, with the corrections included.

What's a Starter Kit?

The Starter Kit is a disk with most of the programs (previously published in COMPUTIST) that you need to get started with disk "snooping" and "converting". This includes SUPER IOB v1.5 (with STANDARD.CON, FAST.CON, SWAP.CON and NEWSWAP.CON), CAPTURE (a routine to convert Applesoft controllers into EXECutable text files), DISKEDIT (for direct disk viewing and editing), the NIBBLER (for viewing raw data from the disk in nibblized form), the CORE DISK SEARCHER (to find byte patterns on the disk), the DOS ALTERER (to make custom DOS changes) and CHECKSOFT and CHECKBIN (to generate the checksums that we print so you can compare and check for errors in your typing).

Where's my Starter Kit?

If you are a new subscriber, you will receive the Starter Kit automatically. If you are renewing and have never received a Starter Kit, ask for the Kit at the same time that you send your renewal and it will be sent to you at no extra charge.

Starter Kit for $2

If you are a current subscriber but you're not ready to renew, then send two dollars and we'll put the Starter Kit in a bubble pack mailer and sent it to you. You do have to be a subscriber. We'll let everyone know when there are significant updates or changes to the the Starter Kit.

BUGS

COMPUTIST #54, page 21

See Thomas Purifoy on page 25

COMPUTIST #57

Table Of Contents

July 1988

Softkeys:

A.P.T.s:

Playing Tips:

IBM Softkeys:

Features:

Notes:

Readers Data

For serious users of Apple computers: II, II+, //c, //e, IIgs, Macintosh

New COMPUTIST readers using Apple IIs are advised to read this page carefully to avoid frustration when attempting to follow a softkey or 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 backup copy can usually be copied by the normal copy programs (for example: *COPYA*, on their *DOS 3.3 System Master* disk).

Commands and control keys

In any article appearing in COMPUTIST, commands which a reader is required to perform are set apart by being in boldface and on a separate line. The RETURN key must be pressed at the end of every such command unless otherwise specified. Control characters are specially boxed. An example of both is:

6 CTRL P

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

Other special combination keypresses include CTRL RESET or CTRL RESET. In the former, press and hold down CTRL then press RESET. In the latter, press and hold down both CTRL and then press RESET.

Special requirements

Special prerequisites for COMPUTIST articles, programs and softkeys are usually listed at the start under:

■ **Requirements:**

Software recommendations

- Applesoft program editor such as *Global Program Line Editor (GPLE)*.
- Sector-editor such as *DiskEdit* (in the *COMPUTIST Starter Kit*) or *ZAP* from *Bag of Tricks*.
- Disk-search utility such as *The Inspector*, the *CIA* or the *Core Disk Searcher* (in the *COMPUTIST Starter Kit*).
- Assembler such as the *S-C Assembler* from S-C software or *Merlin/Big Mac*.
- Bit-copy program such as *Copy II Plus*, *Locksmith* or *Essential Data Duplicator* (EDD).
- 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 and Controllers

This powerful deprotection utility (in the *COMPUTIST Starter Kit*) and its various Controllers are used in many softkeys. (It is also 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 II+, //e, 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 II+, 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, //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:

- ***Apple II Reference Manual***
- ***DOS 3.3 manual***
- ***Beneath Apple DOS***, by Don Worth and Pieter Lechner, from Quality Software
- ***Assembly Language For The Applesoft Programmer***, by Roy Meyers and C.W. Finley, from Addison Wesley

Keying in Applesoft programs:

BASIC programs are printed 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 typed appear in COMPUTIST as delta characters (^). All other spaces are there for easier reading. NOTE: If you want your checksums (See **Computing checksums**) to match up, you must key ONLY the ^spaces after DATA statements.

Keying In Hexdumps

Machine language programs are printed in COMPUTIST as hexdumps, sometimes also as source code. 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 ($ and four digits) at the end of each line. 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:

3D0G

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

The source code is printed to help explain a program's operation. To key it in, you will need the *S-C Assembler* or you will have to translate pieces of the source code into something your assembler will understand (see table of *S-C Assembler* directives in COMPUTIST #17.

Computing checksums

Checksums are 4-digit hexadecimal numbers which tell if you typed 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 accompanying program listings and hexdumps 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*
BRUN CHECKSOFT

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

CHECKBIN instructions:

CALL -151
BLOAD *filename*

Install *CHECKBIN* at an out of the way place

BRUN CHECKBIN, A$6000

Get the checksums by typing the Starting address, a period and the Ending address of the file followed by a CTRL Y.

***SSS.EEE* CTRL Y**

Correct the lines at which the checksums differ.

. . . . RDEXed

EXchange

etc... who want all their software backed up and COPYA-able

when writing a letter to...

Apple RDEXed

RDEX stands for:

Reader's Data EXchange

That means that when you send in articles, softkeys, APTs, etc., you are submitting them for FREE publication in this magazine. **RDEX does NOT purchase submissions nor do we verify data submitted by readers.** We will print it and it is the responsibility of the readers to send in responses when anything is wrong.

- Remember that your letters or parts of them may be used in RDEX even if not addressed to the Apple-RDEX editor. Correspondence that gets published may be edited for clarity, grammer and space requirements .

- Because of the great number of letters we receive and the ephemeral and unpredictable appearance of our part-time staff, any response to your queries will appear only in Apple-RDEX, so it would be more appropriate for you to present technical questions to the readers and ask for their responses which will then be placed in the Apple-RDEX.

Send your articles and letters on

DOS 3.3 standard text files

When we get your letter-article in a standard DOS3.3 text file, it is *immediately* uploaded into the most current RDEX file. Conventional letters must be typed in by us...when we have the time.

- Address your letters, articles, to:

COMPUTIST
Apple RDEX Editor
PO Box 110846-K
Tacoma, WA 98411

Bill Wilson

Slightly embarrassed. That's the only way to describe my feelings after reading William Green's crack for *Game Maker*. In a crack I developed, it took 31 steps to do what Mr. Green has accomplished in a five-byte sector edit. Oh well.

However, I have sent in this crack purely for the educational value of it. If anyone wishes to deprotect *Game Maker* (*GM*), I heartily suggest they use Mr. Green's crack in COMPUTIST #5Ø. In the following information, you may find various techniques and processes of value for future cracking attempts.

Softkey for...

Game Maker

Activision

■ Requirements

□ *Game Maker* disk
□ Blank disk
□ A formatted DOS 3.3 slave disk
□ A sector editor
□ Apple IIe with at least 64K
□ One disk drive

Game Maker (GM) is a useful utility for creating customized computer games. Using the various editors of *GM*, one can design background hi-res graphics screens, animated sprites, sound effects, musical scores, and program control code. Once a satisfactory game has been designed, it may be saved to disk or a special disk may be created that automatically boots the game. *GM*, however, suffers from a traditional disease that has affected all Activision software. It is copy-protected.

For those of you who are not interested in the mechanics of the crack, you can find the quick fix for *GM* in the cookbook instructions.

On the surface, *GM* is not copy-protected at all. *COPYA*, the *Locksmith Fast Copier*, or *Copy II Plus*' Copy Disk Option will work fine on *GM*. However, when the copy is booted, the boot will halt and the user will find himself with a locked-up computer.

When this occurred on my Apple, I used my *Wildcard* software and firmware to see where the lockup (actually an infinite loop) occurred. At $824Ø, I found the instruction JMP $824Ø. This is what was causing the computer to lock up. My immediate thought was to locate the routine on disk and neutralize it with a well-aimed branch to an instruction to continue loading. I found the code for $8ØØØ-82FF on track $21, sectors $Ø5, $Ø6, and $Ø7. (I used the "scan for hex bytes" option of *Copy II Plus* to locate the bytes 4C 4Ø 82, thus leading me to the infinite loop.) I tried to put the bytes EA EA EA (no-operation instructions) in the place of the loop. Booting the disk, however, produced the same result as before. The boot would cease and nothing would happen.

Trying various branches proved to be of no avail and only led to the same result with the disk refusing to complete a boot cycle. I was now convinced there was a fairly sophisticated copy-protection routine operating during the boot. It became obvious to me the program needed something loaded by the routine at $8ØØØ to continue loading the program and start the utility. I pulled the code for $8ØØØ-82FF from the disk with *Copy II Plus* and listed it to my printer.

Using a stripped down version of Don Lancaster's "tearing method", I highlighted all of the jumps (JMP) and jump to subroutines (JSR) in the code. By doing this, along with marking the return from subroutines (RTS), one can follow the general flow of a program without necessarily understanding what it is doing at every step of the way. Analyzing the code brought out these facts: One, the routine exited to $ADØØ whether or not the disk check was OK. Two, if the disk check was not OK, the bytes 4C 4Ø 82 (JMP $824Ø) were inserted at $ADØØ prior to jumping there.

At this point, the user should make a *COPYA* copy of *GM*, as it will be used later in the crack.

Since the disk check produced something that I needed, to boot the disk completely, I decided the best way to capture the code at $ADØØ would be through a boot trace. A boot trace is a way of controlling and redirecting a disk's boot cycle so that you are in control of it at all times. To do this, one must start with BootØ, the code in the disk controller card at $C6ØØ-C6FF. I used these commands to move BootØ down to page $96ØØ so it could be modified.

```
CALL-151                         enter the monitor
96ØØ<C6ØØ.C6FFM           move BootØ to RAM
96FA:98 N 98Ø1:4C 59 FF        redirect boot
96ØØG                              boot the disk
CØE8                        turn off disk drive
```

Now, we have captured Boot1, which is sitting at $Ø8ØØ-Ø8FF. We will move it to

$98ØØ so it can be modified and worked with to meet our wishes.

```
98ØØ<8ØØ.8FFM          move the code
98ØE:9Ø                adjust so it uses our code at
                       $96ØØ not the ROM at $C6ØØ
984A:4C 59 FF          jump to monitor after Boot 2 loads
96ØØG                  boot the disk
CØE8                   turn off disk drive
```

Boot2 is now captured and sits at $B6ØØ-BFFF. Again, it will be moved so we can modify it.

```
46ØØ<B6ØØ.BFFFM        move the code
984A:4C ØØ 47          adjust Boot1 to jump to
                       the new location of boot2
4741:4C 59 FF          add jump to monitor
96ØØG                  boot the disk
```

Boot3 (the code at $8ØØØ-82FF) has now been loaded at $8ØØØ. Again, a move and modification is in order to further control the boot.

```
3ØØØ<8ØØØ.82FFM        move the code
4741:4C ØØ 3Ø          adjust Boot 2 to jump
                       to the new location of Boot 3
3Ø3C:3Ø                adjust Boot 3 to work
3ØF2:31                at its new location
313B:32
3227:32
3242:32
3269:4C 59 FF          add jump to monitor
96ØØG                  boot the disk
```

A listing from $ADØØ should now reveal rational code. This routine is what actually loads *GM* and executes the utility. At this point, the *Game Maker* disk should be removed from the drive and a DOS 3.3 formatted disk inserted. The DOS 3.3 disk should be a slave disk with the DOS image on it and NOT be a DOS 3.3 master disk. (If you are not familiar with this terminology, a slave disk is simply a disk formatted with the DOS 3.3 INIT command. A DOS 3.3 master disk is made by using the Master Create utility.) Now, execute these commands from the monitor:

```
1DØØ<ADØØ.B5FFM        move code down
96ØØG                  boot the slave disk
BSAVE GM.LOADER, A$1DØØ, L$8FF
```

At this point, the usual thing to do would be to find room for the file on the copied *GM* disk and to make GM.LOADER the "Hello" file of the disk. That cannot be done here since *GM* does not use all of DOS 3.3 and since most of the disk is filled with program code and data. What we will have to do is find a place for the file and adjust Boot 1 to load our GM.LOADER instead of the protection routine. A scan of the disk with *Copy II Plus*' "Track/Sector Map" utility revealed no free space large enough to store the file. However, there are only a few large files on the disk. This means the catalog track will have some free sectors. As it turns out, we can store the file GM.LOADER on track $11, sectors $Ø3-B.

Here is what the program to do this looks like.

```
Ø9ØØ A2 6Ø    LDX #$6Ø  X-reg = slot 6
Ø9Ø2 A9 Ø9    LDA #$Ø9  write 9 pages
Ø9Ø4 8D EØ B7 STA $B7EØ
Ø9Ø7 A9 11    LDA #$11  use track $11
Ø9Ø9 8D 15 B7 STA $B715
Ø9ØC A9 ØB    LDA #$ØB  start at sct $ØB go down
Ø9ØE 8D 1A B7 STA $B71A
Ø911 A9 26    LDA #$26  start page of memory + 1
                        that the RWTS will
Ø913 8D E7 B7 STA $B7E7 work down from
Ø916 4C ØØ B7 JMP $B7ØØ write the data to disk
```

Insert the *COPYA* version of *GM* into drive 1 enter these commands:

```
CALL-151
B726:Ø2                     RWTS write cmnd
B7EB:ØØ                     any volume # is OK
Ø9ØØ:A2 6Ø A9 Ø9 8D EØ B7 A9
Ø9Ø8:11 8D 15 B7 A9 ØB 8D 1A
Ø91Ø:B7 A9 26 8D E7 B7 4C ØØ B7
9ØØG
```

GM.LOADER will be written to the appropriate sectors of track $11. We have the captured loader on disk now, but the disk still won't boot. BootØ must be modified to load GM.LOADER after it has loaded Boot1, and Boot1 must be modified to jump to GM.LOADER instead of jumping to the protection routine. We also need to clean up the catalog track so we won't get a bunch of garbage and disk errors when we catalog the disk. Get your sector editor out and make these edits to the *COPYA* version of *GM*:

Trk	Sct	Byte(s)	From	To
$ØØ	$ØØ	$4A	?	4C 8Ø Ø8
		$8Ø	?	A9 Ø9 8D EØ B7 A9 11 8D 15 B7 A9 ØB 8D 1A B7 A9 B6 8D E7 B7 4C ØØ B7
	$Ø1	$15	?	11
		$1A	?	ØB
		$1F	?	B5
		$EØ	?	Ø9
$11	$ØC	$Ø1	?	ØØ ØØ

With these edits, you now have a *COPYA*, fully functional version of *GM*. The back side of *GM* may be copied with *COPYA*.

Here are the cookbook instructions for deprotecting *Game-Maker*:

1 Make a *COPYA* copy of *Game-Maker*.

```
CALL-151
96ØØ<C6ØØ.C6FFM
96FA:98 N 98Ø1:4C 59 FF
```

2 Insert the *Game-Maker* original disk into drive one.

```
96ØØG
CØE8
98ØØ<8ØØ.8FFM
98ØE:9Ø N 984A:4C 59 FF
96ØØG
CØE8
46ØØ<B6ØØ.BFFFM
984A:4C ØØ 47
4741:4C 59 FF
96ØØG
3ØØØ<8ØØØ.82FFM
4741:4C ØØ 3Ø
3Ø3C:3Ø N 3ØF2:31 N 313B:32
3227:32 N 3242:32
3269:4C 59 FF
96ØØG
1DØØ<ADØØ.B5FFM
```

3 Insert a DOS 3.3 formatted slave disk and save the loader to disk in case you glitch the softkey.

```
C6ØØG
BSAVE GM.LOADER
```

4 Insert the *COPYA* version of *Game-Maker*.

```
CALL-151
B726:Ø2 N B7EB:ØØ
Ø9ØØ:A2 6Ø A9 Ø9 8D EØ B7 A9
Ø9Ø8:11 8D 15 B7 A9 ØB 8D 1A
Ø91Ø:B7 A9 26 8D E7 B7 4C ØØ
Ø918:B7
9ØØG
```

5 Make the following sector edits to *COPYA* version of *Game-Maker*:

Trk	Sct	Byte(s)	From	To
$ØØ	$ØØ	$4A	?	4C 8Ø Ø8
		$8Ø	?	A9 Ø9 8D EØ B7 A9 11 8D 15 B7 A9 ØB 8D 1A B7 A9 B6 8D E7 B7 4C ØØ B7
$ØØ	$Ø1	$15	?	11
		$1A	?	ØB
		$1F	?	B5
		$EØ	?	Ø9
$11	$ØC	$Ø1	?	ØØ ØØ

6 Copy the back side of *Game-Maker* with *COPYA* onto another disk and you are done.

Whew! William Green's five-byte sector edit sure beats doing all of that. To understand why the five-byte sector edit worked, it is necessary to examine the protection on *Game-Maker*. The loader at $ADØØ is necessary to load the main program from the disk. This loader is encrypted on disk and the key to decrypt it is found through a nibble count. No original disk, no decrypted loader.

By using the boot-trace method to coax the decrypted loader from the disk, I captured and used it to load the rest of the program. This can be considered the "infantry wade-in and take no prisoners" brute assault on the nibble count to access the protected information.

Let's look at a documented version of Mr. Green's crack, on track $21, sector $Ø5, starting at byte $54:

```
A9 FF  LDA #$FF  $FF is the decrypt key
18     CLC       clear carry to force branch
9Ø 3D  BCC ??    branch to the decrypt
                 routine
```

Mr. Green's tactic was simply to discover the decryption key by whatever means, load the accumulator with it, and branch to the decryption routine. Simple and highly effective. To be candid, I had considered this, but could not decide on where to branch to into the nibble count routine.

I hope this long softkey will provide an example of in-depth disk analysis to beginning disk-busters who read COMPUTIST. It certainly shows how NOT to crack *Game-Maker*! Yet, it also shows there is more than one way to defeat a nibble count, albeit a tedious way.

Softkey Addendum for...

Master Diagnostics IIe

Nikrom

The softkey for *Master Diagnostics IIe* (MDIIE) as printed in COMPUTIST #52 is not complete, and this is my fault. The softkeyed disk will work fine unless the disk drive test routines are invoked.

The reason these routines print garbage on the screen and bomb out is due to a DOS change made by Nikrom. At locations $BA69 (47721) and $BA7D (47741), Nikrom inserted routines that swap disk data marks. They had to; otherwise the original MDIIE disk would never be able to read normally formatted disks to conduct the tests. On a disk copied by *Super IOB*, the calls to these locations become logic bombs that stop the Apple cold.

There are several approaches to remedy this problem. One, the locations $BA69 and $BA7D could be changed to 6Ø (Return from Subroutine) on the DOS of the SIOB'd disk. This is not desirable because late versions of DOS 3.3 use these locations for their own purposes and the person cracking the disk should not have to worry about the 3.3 version. Two, all of the Applesoft programs these calls appear in could be loaded, edited to remove the calls, and saved to disk. This would be tedious. Three, the calls can be edited on disk to direct the call to a location in DOS that already has a RTS instruction. This is the best method I found.

The code to call these routines was CALL 47741 and CALL 47721. Since "477" was common to both of these instructions, I used *Copy II Plus* to scan the disk for the hex bytes 34 37 37, which is the way Applesoft stored these three digits on disk in ASCII. This quickly located the sectors in question. I changed the last two digits of 47741 and 47721 from 41 and 21 to 33. There is an RTS instruction at location 47733 ($BA75). Here are the edits to make on the SIOB'd version of MDIIE:

Trk	Sct	Byte(s)	From	To
$Ø5	$ØE	$5D	?	33 33
$Ø6	$Ø5	$B1	?	33 33
	$Ø8	$9D	?	33 33
	$ØE	$44	?	33 33
$ØE	$Ø7	$6Ø	?	33 33
	$ØE	$2F	?	33 33
$1Ø	$Ø6	$BE	?	33 33
		$DD	?	33 33
	$ØE	$21	?	33 33

Editing these nine sectors will make the drive routines fully functional. Sorry about any problems this oversight may have caused! One final note. In case you have a very early version of DOS 3.3, the RTS instruction at $BA75 may not exist. In this case, you should redo the softkey, this time including a POKE 47733,96 after the POKE 4Ø514,52 prior to the copy disk initialization. This new POKE ensures an RTS instruction will be present at $BA75.

Joe Walters

Softkey for...

Microzine #24

Scholastic

A REM statement at line number zero of the AppleSoft program called HELLO contains a D followed by FP. This causes the program to be erased in memory if you try to list it. This is easily defeated by removing the offending statement since it performs no useful function (from our standpoint).

Secondly, there is a nibble count program called CP.OPTIONS that is called from line three of the same HELLO program. That too can be defeated by simply REMing it out since I could not detect any other program making use of any output from the nibble count program.

1 Insert side one of Microzine #24 in drive 1.

```
LOAD HELLO
Ø REM
3 POKE (767),2 : REM PRINT CHR$(4)"BRUN
    CP.OPTIONS"
SAVE HELLO
```

The disk is deprotected.

The remainder of this article shows a commented disassembly of CP.OPTIONS so you can become familiar with one type of nibble count program.

Notes:

1. When a RTS (Return from subroutine) instruction is executed the address stored on the stack has 1 added to it in order to obtain the true return address. So, when $C5FF is stored on the stack and a RTS occurs we will return to $C6ØØ.
2. The disk controller card is the one that actually reads the data from the disk. Location $CØ8C is the output of the card. $CØ8C is made zero and then the byte is read in a bit at a time starting with disk bit 7 which is initially placed in bit Ø of $CØ8C. As each succeeding bit is read in the preceding bit(s) are shifted left one place and the new bit is placed in bit Ø. Since all disk bytes must have bit 7 = 1, it follows that when $CØ8C shows bit 7 set the byte is complete.

```
RWTS   = $Ø3D9 Vector address of RWTS
GETIOB = $Ø3E3 Get RWTS parm list address
MTROFF = $CØ88 Drive motor off
MTRON  = $CØ89 Drive motor on
STROBE = $CØ8C Minus when a byte is all in

9ØØØ 2Ø E3 Ø3 JSR GETIOB
9ØØ3 85 FB    STA $FB      MSB parm list
9ØØ5 84 FA    STY $FA      LSB parm list
9ØØ7 A9 C5    LDA #$C5     MSB of $C5FF to stack. See
9ØØ9 48       PHA          below for how $FF makes it
9ØØA A9 ØØ    LDA #$ØØ     Init loop counter
9ØØC 85 FC    STA $FC
9ØØE A2 Ø3    LDX #$Ø3
9Ø1Ø BC 35 9Ø LDY $9Ø35,X Set parmlist with zeroes
9Ø13 91 FA    STA ($FA),Y at indices from table.
9Ø15 CA       DEX          X = 3, 2, 1. Ø, $FF
9Ø16 1Ø F8    BPL $9Ø1Ø    Fall through when X = $FF
9Ø18 8A       TXA          $FF from X -> A -> stack
9Ø19 48       PHA          Stack now = $C5FF
9Ø1A 2Ø 28 9Ø JSR $9Ø28    Seek to track zero
9Ø1D AØ Ø1    LDY #$Ø1
9Ø1F B1 FA    LDA ($FA),Y Get slot*16 from
                           parmlist
9Ø21 AA       TAX          Move slot*16 to X
9Ø22 2Ø 39 9Ø JSR $9Ø39    Nibble away at it
9Ø25 68       PLA          Remove $C5FF from stack
9Ø26 68       PLA
9Ø27 6Ø       RTS          Return success
9Ø28 2Ø E3 Ø3 JSR GETIOB
9Ø2B 2Ø D9 Ø3 JSR RWTS
9Ø2E A9 ØØ    LDA #$ØØ
9Ø3Ø 85 48    STA $48
9Ø32 BØ 52    BCS $9Ø86    Error on seek = panic time!
9Ø34 6Ø       RTS
```

The following hex data table is used to get the drive head to Track zero.

```
Ø8 = LSB user's buffer    Ø4 = Track
ØC = Command (seek)    Ø3 = Vol.

9Ø35 Ø8 Ø4 ØC HEX Ø8Ø4ØCØ3
9Ø38 Ø3       HEX Ø3
```

Here is where the nibble count begins.

```
9Ø39 BD 89 CØ LDA MTRON,X Motor (back) on
9Ø3C A9 56    LDA #$56    Loop counter for looking
9Ø3E 85 FD    STA $FD     for $FB's
9Ø4Ø A9 Ø8    LDA #$Ø8
9Ø42 C6 FC    DEC $FC     $FF, $FE, ....Ø, $FF,...
9Ø44 DØ Ø4    BNE H9Ø4A
9Ø46 C6 FD    DEC $FD     $56, $55, ....Ø
9Ø48 FØ 3C    BEQ H9Ø86   Exit failure
```

When a disk is initialized it is with an address header of D5 AA 96, a data header of D5 AA AD and a data block (96 96 96...) full of zeros. This is repeated for 16 sectors. (ADZ ADZ ADZ ADZ... 16 times)

The number of zeros is enough to reserve room for the future data. The nibble counts usually hide after the data before the next Address Header. We will be looking for the sequence of FB FF ff FF ff FF ff, Where the FF's are 1Ø bit bytes (ØØ 1111 1111) and the ff's are normal 8 bit bytes (1111 1111). The one shot read at $9Ø57 will yield something less than $Ø8 for 1Ø bit nibbles and something greater for 8 bit nibbles. (Hard to use Copy II Plus or Locksmith to scan for FB's followed by the above.)

The number(s) in the comment fields give the cycles for that instruction. In the case of numbers like 2/3, the first is for fall through, the second is when the branch is taken.

Each disk byte takes 32µs to read which is 4µs per bit, Since the processor runs at 1µs, 32µs is 32 machine cycles. Thus, each bit on the disk takes 4 machine cycles.

For 8 bit bytes (ff) we have:

```
 µs 4 4 4 4  4 4 4 4 = 32µs
bits 1 1 1 1  1 1 1 1
```

For 1Ø bit bytes (FF) we have:

```
 µs 4 4  4 4 4 4  4 4 4 4 = 4Øµs
bits Ø Ø  1 1 1 1  1 1 1 1
```

Ten bit bytes are also known as self synch bytes and are explained at great length in that ever popular tome, "Beneath Apple DOS."

The disk controller card starts reading a byte off the disk by first zeroing $CØ8C, then the next byte is shifted in from the right, a bit at a time, every 4µs. Since it takes 16µs to 2Øµs to go from $9Ø4A through $9Ø57 it is clear that Y is greater than $Ø8 for 8 bit bytes: 1111 (16) or 11111 (32).

It follows that Y is less than $Ø8 for 1Ø bit bytes: ØØ11 (3) or ØØ111 (7). (Fast disk drives could mess this up)

If the above is unclear then consider both the code below and the note above at length.

```
9Ø4A BC 8C CØ LDY STROBE,X4 - Wait for a disk byte
9Ø4D 1Ø FB    BPL H9Ø4A   2/3 - B/not yet
9Ø4F CØ FB    CPY #$FB    2 - Was it $FB
9Ø51 DØ ED    BNE H9Ø4Ø   2/3 - No/keep looking
9Ø53 FØ ØØ    BEQ H9Ø55   2/3 - Yes Xfer & waste cycle
9Ø55 EA       NOP         2 - Waste a couple more
9Ø56 EA       NOP         2
```

Note that the LDY STROBE,X below is NOT followed by a BPL This means we will take whatever has been seen by the disk controller card at this point.

```
9Ø57 BC 8C CØ LDY STROBE,X4 - Get partial nibble
9Ø5A CØ Ø8    CPY #$Ø8    2 - Set carry if Y >= $Ø8
```

"A" starts out as $Ø8 & each time through is changed:

```
ØØØØ1ØØØ, ØØØ1ØØØØ, ØØ1ØØØØØ, Ø1ØØØØØØ. 1ØØØØØØØ
```

Last time through the carry is set & we go for final nibble. Note that "A" will (and is) something different if the carry was set.

This is the real sequence:

```
ØØØØ1ØØØ.ØØØ1ØØØØ.ØØ1ØØØØ1.Ø1ØØØØ1Ø.1ØØØØ1Ø1,ØØØØ1Ø1Ø
```

Init	1st	2nd	3rd	4th	Last ROL
	FF	ff	FF	ff	A = $ØA

```
9Ø5C 2A       ROL         2
9Ø5D BØ ØB    BCS H9Ø6A   2/3 If set, look for final
9Ø5F BC 8C CØ LDY STROBE,X 4
9Ø62 1Ø FB    BPL H9Ø5F   2/3 Wait partial to cmplt
9Ø64 CØ FF    CPY #$FF    2 Was it = $FF (or ff)
9Ø66 DØ D8    BNE H9Ø4Ø   2 No Start over
9Ø68 FØ EB    BEQ H9Ø55   3 Always Xfer & Get
                          partial
9Ø6A BC 8C CØ LDY STROBE,X Look for byte
9Ø6D 1Ø FB    BPL H9Ø6A
9Ø6F 84 FC    STY $FC     $FF - trust me
9Ø71 C9 ØA    CMP #$ØA    Was A = $ØA from above?
9Ø73 DØ CB    BNE H9Ø4Ø   No, start over
9Ø75 BD 8C CØ LDA STROBE.X Another $FF - trust again
9Ø78 1Ø FB    BPL H9Ø75
```

Misdirection. A = $FF and this just keeps it so.

```
9Ø7A 38       SEC         Going to set LSB of A
9Ø7B 2A       ROL         Keep A = $FF
```

Think about things for a second:

1. The only value A can have before the EOR is $FF if it is going to be zero so the BNE doesn't take.

2. A must = $FF before the AND and $FC must contain $FF since these are the only values that when ANDed together produce $FF.

```
9Ø7C 25 FC    AND $FC     $FF & $FF = $FF
9Ø7E 49 FF    EOR #$FF    $FF xor $FF = $ØØ
9Ø8Ø DØ Ø4    BNE H9Ø86   B/if wrong & die
9Ø82 DD 88 CØ CMP MTROFF.X Shut of disk drive
9Ø85 6Ø       RTS         Return Success
9Ø86 A8       TAY
9Ø87 DD 88 CØ CMP MTROFF.X Off motor
9Ø8A 68       PLA         Pull return from JSR H9Ø39
9Ø8B 68       PLA         off so we return to $C6ØØ
9Ø8C 99 ØØ 9Ø STA H9ØØØ.Y Wipe this routine
9Ø8F C8       INY
9Ø9Ø CØ 8B    CPY #$8B
9Ø92 DØ F8    BNE H9Ø8C
9Ø94 6Ø       RTS
```

Phil Goetz

If anyone would like to contact me, my BITNet address is PGOETZ@LOYVAX.bitnet

Electronic Arts Protection Language

I was boot-code tracing *Amnesia*, and I came to $37ØØ. A protection routine starting at $37Ø4 evidently checked for one sector of nothing but $AFs, but I didn't see where it was called. The tracing became very complicated and indirect, but I noticed that I kept circling back to $38C3. Then I realized that $38C3 was the equivalent of Applesoft's GETCHR for EOA's own interpretive language. After much tracing and disassembling, I decoded this language, which I call EOAPL. It is basically a small subset of 65Ø2 machine language, and it is of course much slower, so its only purpose is to confuse crackers.

To see this code, boot *Amnesia*. When it asks you PLAY A SAVED GAME? insert a normal disk into the drive and press **RESET**. All the code is left unaltered.

In *Amnesia*, $37ØØ jumps to $376A:

```
376A JSR $38E1
376D JSR $38B4
377Ø JMP $376A
```

The "JSR $38B4" is the call to the interpreter. It stores the address 3 bytes beyond the next instruction in the interpreter's program counter (in this case, $3773) and falls through to $38C3, which starts running the program at $3773.

$38C3 is the crucial routine to look for:

```
38C3 B1 52    LDA ($52),Y Get opcode
38C5 C8       INY         Advance program counter
38C6 DØ Ø2    BNE $38CA
38C8 E6 53    INC $53
38CA AA       TAX
38CB BD D4 38 LDA $38D4,X Get low byte of address
                          which executes this
                          instruction
38CE 8D EØ 39 STA $39EØ   Store it in the JMP
38D1 4C DF 39 JMP $39DF   Execute; routine will
                          return via a JMP $38C3
39DF 4C xx 39 JMP $39xx
```

Each instruction consists of a 1-byte opcode followed by a Ø, 1, or 2 byte parameter.

The opcodes are not encoded. Each 1 byte parameter is EORed with $4C. Every 2 byte parameter is EORed with $D9Ø3 (the low byte is EORed with $Ø3 and the high byte with $D9).

EOAPL uses $56 as the accumulator and $53,52 as the program counter. $5Ø-5F are used for various things.

The commands are:

Op.	Len	Addr	Mnemonic	Function
Ø	3	39ØØ	GOTO addr	
1	3	3927	JSRV addr	Call a machine language subroutine with $56 as a value parameter: On entry A ← $56, on exit $56 ← A.
2	3	3955	GONE addr	Go to addr if $56 > Ø
3	2	395F	LD56 num	
4	3	396D	LD56 addr	
5	3	398A	GSUB addr	
6	3	39AD	ST56 addr	
7	2	39B9	S56- num	$56 ← $56 – num
8	1	39A2	RTRN	Return from subroutine
9	3	394F	JSR addr	Call a machine language subroutine
A	3	39CE	INC addr	INC addr : $56 ← (addr)
B	1	3926	RTS	Exit program & return to machine language caller
C	3	3979	S56+ addr	$56 ← $56 + (addr)

So, for example, a routine to print "4C" would look like this:

```
Ø3ØØ 2Ø B4 38   JSR $38B4
Ø3Ø3 6Ø         RTS
Ø3Ø4 ØØ         BRK
Ø3Ø5 ØØ         BRK
```

Disassembled EOAPL.

```
Ø3Ø6 Ø3 ØØ      LD56#$4C
Ø3Ø8 Ø1 D9 24   JSRV$FDDA
Ø3ØB ØB         RTS
```

About $BØ bytes are in EOAPL - too many to disassemble by hand. So I rewrote the monitor disassembler ($F882-FA3F, enter at $F8DØ) to disassemble EOAPL. Both the source and hex code is included here. The source is written for the Apple Toolkit Assembler. To enter the hex code, follow these steps:

[1] Enter the monitor.

CALL -151

[2] Enter the EOAPL disassembler hex code. Remember: the numbers with the "$" are checksums, you don't enter those. You compare them with the numbers you get from CHECKBIN to see if you made any errors.

[3] Now save the binary code to disk.

BSAVE EOADIS,A$8ØØ,L$112,V61

There are three simple steps to use the EOAPL disassembler from the monitor. First, put the start address in $Ø1,ØØ, then put the number of lines to disassemble in $Ø2 and, finally, type 8ØØG to call the disassemble.

So to disassemble the code in *Amnesia* at $3773, we type:

Ø:73 37 13
8ØØG

and we get:

```
3773 Ø4 62 EE  LD56 $3761
3776 Ø7 74     S56- #$38
3778 Ø2 8E E1  GONE $388D Crash if $3761 is changed
377B Ø4 61 EE  LD56 $3762
377E Ø7 2C     S56- #$6Ø
378Ø Ø2 8E E1  GONE $388D Same for $3762
3783 Ø4 6B EE  LD56 $3768
3786 Ø7 74     S56- #$38
3788 Ø2 8E E1  GONE $388D Same for $3768
378B Ø5 AB EE  GSUB $37A8 Call $37Ø4 & check for
                          original
378E Ø2 97 EE  GONE $3794 Fail, try again
3791 ØØ A6 EE  GOTO $37A5 OK, exit
3794 Ø5 AB EE  GSUB $37A8
3797 Ø2 9E EE  GONE $379D Fail
379A ØØ A6 EE  GOTO $37A5 OK, exit
379D Ø7 4E     S56- #$Ø2
379F Ø2 8E E1  GONE $388D Crash if $56 wasn't 2 after
                          $37A8
37A2 ØØ 7Ø EE  GOTO $3773 Do it again
37A5 Ø9 A4 E1  JSR  $38A7 Exit to caller
```

Inspection of the disassembly shows us we can bypass the protection by switching the GONE $3794 at $378E with the GOTO $37A5 at $3791. That way it exits OK regardless of what happened, and any checksums are satisfied. Scanning the disk, we find Ø2 97 EE ØØ A6 EE on track $Ø9, sector $Ø2 at bytes $E2-E7. So to crack *Amnesia*, copy all tracks except track $Ø6 and perform this sector edit on disk 1:

Trk	Sct	Byte(s)	From	To
$Ø9	$Ø2	$E2-E4	Ø2 97 EE	ØØ A6 EE
		$E5-E7	ØØ A6 EE	Ø2 97 EE

EOAPL is also used in *The Last Gladiator*, *Archon*, *Archon II: Adept*, and probably many other EOA releases.

EOAPL Disassembler Source

```
        ORG $8ØØ
CODE    EQU 2
FORMAT  EQU $2E
LENGTH  EQU $2F
        LDA Ø
        STA $3A
        LDA 1
        STA $3B
        LDA 2
        STA 6
L1      JSR EOADIS
        LDA LENGTH
        SEC
        ADC $3A
        STA $3A
        BCC L2
        INC $3B
L2      DEC 6
        BNE L1
        RTS
EOADIS  JSR INSDS1
PRNTOP  LDA ($3A),Y
        JSR $FDDA
        LDX #1
PRNTBL  JSR $F94A
        CPY LENGTH
        INY
        BCC PRNTOP
        LDX #3
        CPY #4
        BCC PRNTBL
;PRINT MNEMONIC
        LDA CODE
        ASL A
        ASL A
        TAY
        LDX #4
PMN     LDA MTAB,Y
        JSR $FDED
        INY
        DEX
        BNE PMN
        JSR $F948
        LDY LENGTH
        LDX #6
PRADR1  CPX #3
        BEQ PRADR5
PRADR2  ASL FORMAT
        BCC PRADR3
        LDA $F9B3,X
        JSR $FDED
        LDA $F9B9,X
        BEQ PRADR3
        JSR $FDED
PRADR3  DEX
        BNE PRADR1
        RTS
PRADR4  DEY
        BMI PRADR2
        STX 3
        LDX LENGTH
        CPX #2
        BCS ED9Ø3
        EOR #$4C
        JMP PR45
ED9Ø3   EOR ETAB,Y
PR45    JSR $FDDA
        LDX 3
PRADR5  LDA FORMAT
        CMP #$E8
        LDA ($3A),Y
        BCC PRADR4
        JSR $F956
        TAX
        INX
        BNE PRNTYX
        INY
PRNTYX  TYA
        JSR $FDDA
        TXA
        JMP $FDDA
RTS     RTS
ETAB    DFB $3,$D9
LENTAB  DFB 2,2,2,1,2,2,2,1,Ø,2,2,Ø,2,Ø
FMTTAB  DFB $8Ø,$8Ø,$8Ø,$2Ø,$8Ø,$8Ø,$8Ø,
            $2Ø,Ø,$8Ø,$8Ø,Ø,$8Ø,Ø
```

```
MTAB    ASC /GOTOJSRVGONELD56LD56GSUBST56
            S56-RTRNJSR INC RTS S56+????/
INSDS1  LDX $3A
        LDY $3B
        JSR $FD96
        JSR $F948
INSDS2  LDA ($3A,X)
        CMP #$D                     LEGAL OPCODE?
        BCC GETFMT                            YES
        LDA #$D                              ????
GETFMT  TAY
        STA CODE
        LDA LENTAB,Y
        STA LENGTH
        LDA FMTTAB,Y
        STA FORMAT
        LDY #Ø
        RTS
```

EOAPL Disassembler hex dump

```
Ø8ØØ: A5 ØØ 85 3A A5 Ø1 85 3B          $6CBE
Ø8Ø8: A9 16 85 Ø6 2Ø 1F Ø8 A5          $82F1
Ø81Ø: 2F 38 65 3A 85 3A 9Ø Ø2          $4FC9
Ø818: E6 3B C6 Ø6 DØ EE 6Ø 2Ø          $78B4
Ø82Ø: FØ Ø8 B1 3A 2Ø DA FD A2          $7E59
Ø828: Ø1 2Ø 4A F9 C4 2F C8 9Ø          $55BF
Ø83Ø: F1 A2 Ø3 CØ Ø4 9Ø F2 A5          $DC5Ø
Ø838: Ø2 ØA ØA A8 A2 Ø4 B9 B8          $734C
Ø84Ø: Ø8 2Ø ED FD C8 CA DØ F6          $F831
Ø848: 2Ø 48 F9 A4 2F A2 Ø6 EØ          $C68C

Ø85Ø: Ø3 FØ 2E Ø6 2E 9Ø ØE BD          $EB14
Ø858: B3 F9 2Ø ED FD BD B9 F9          $A669
Ø86Ø: FØ Ø3 2Ø ED FD CA DØ E7          $AF8B
Ø868: 6Ø 88 3Ø E7 86 Ø3 A6 2F          $6666
Ø87Ø: EØ Ø2 BØ Ø5 49 4C 4C 7C          $F1AC
Ø878: Ø8 59 9A Ø8 2Ø DA FD A6          $FBBB
Ø88Ø: Ø3 A5 2E C9 E8 B1 3A 9Ø          $A9DØ
Ø888: EØ 2Ø 56 F9 AA E8 DØ Ø1          $Ø7EE
Ø89Ø: C8 98 2Ø DA FD 8A 4C DA          $DB2E
Ø898: FD 6Ø Ø3 D9 Ø2 Ø2 Ø2 Ø1          $1283

Ø8AØ: Ø2 Ø2 Ø2 Ø1 ØØ Ø2 Ø2 ØØ          $AA8F
Ø8A8: Ø2 ØØ 8Ø 8Ø 8Ø 2Ø 8Ø 8Ø          $9B6F
Ø8BØ: 8Ø 2Ø ØØ 8Ø 8Ø ØØ 8Ø ØØ          $ABØF
Ø8B8: C7 CF D4 CF CA D3 D2 D6          $1FD2
Ø8CØ: C7 CF CE C5 CC C4 B5 B6          $A3ØE
Ø8C8: CC C4 B5 B6 C7 D3 D5 C2          $C32C
Ø8DØ: D3 D4 B5 B6 D3 B5 B6 AD          $9EØC
Ø8D8: D2 D4 D2 CE CA D3 D2 AØ          $511A
Ø8EØ: C9 CE C3 AØ D2 D4 D3 AØ          $F4D2
Ø8E8: D3 B5 B6 AB BF BF BF BF          $5Ø48

Ø8FØ: A6 3A A4 3B 2Ø 96 FD 2Ø          $A29A
Ø8F8: 48 F9 A1 3A C9 ØD 9Ø Ø2          $3C8Ø
Ø9ØØ: A9 ØD A8 85 Ø2 B9 9C Ø8          $72E1
Ø9Ø8: 85 2F B9 AÅ Ø8 85 2E AØ          $FFØF
Ø91Ø: ØØ 6Ø                            $FD41
```

Diskscan:

How to find EOR on disk

I was tired of getting a "HARDWARE FAILURE!" no matter what I tried with *F-15 Strike Eagle*. I searched for the string on disk, but it wasn't there. Knowing Microprose encoded with a simple EOR, I wrote *Diskscan* to scan an entire disk for a string which had been EORed with any value.

To use *Diskscan*: Enter the length of the pattern you want to find in $3ØØ, and the pattern from $3Ø1 on. Then **BRUN SCAN**. It will ask you if you want to SCAN (U)P OR (D)OWN? If you choose D (which is usually the best choice), *Diskscan* will scan the disk from track $22, sector $ØF down to track $ØØ, sector $ØØ. More importantly, if it sees the beginning of the string on track $1A, sector $Ø7, it will look for the rest of it on track $1A, sector $Ø6. If you were scanning up, it would look for the rest on track $1A, sector $Ø8.

Every time it finds a match, it will print the track, sector, byte# of the start of the match, and the value which the pattern was EORed with. You may press ESC at any time to halt it.

So to search for HARDWARE I enter:

```
CALL -151
3ØØ:8 C8 C1 D2 C4 D7 C1 D2 C5
BRUN SCAN
D                                for read down
```

If you tried this on the original, you would get:

```
T$22 S$ØF ERROR #$4Ø
T$22 S$ØE ERROR #$4Ø
```

and so on. But since I am using a normal-format copy, after a few minutes *Diskscan* responds with:

```
T$17 S$ØF 65:CØ
T$ØØ S$ØØ
*                    you're still in the monitor
```

This means that C8 C1 D2 C4 D7 C1 D2 C5 EORed with $CØ is on track $17, sector $ØF starting at byte $65.

Note that Applesoft must be resident, because *Diskscan* calls $DB3A to print. Also, if your DOS is non-standard or in an unusual location, you should replace the JMP $A851 with JMP $3DØ. $3DØ will both reconnect DOS and enter BASIC.

Diskscan is written using the *S-C Macro Assembler*. "/LABEL" means the high-byte of LABEL, "#LABEL" is the low-byte. ".BS n" creates n zeroes. Enter the hex dump and save it as "DISKSCAN, A$2222, L$1D2'.

DISKSCAN source code

```
        .LIF
* ...............LOCATE EOR-ENCODED STRING
        .OR $2222
        .TF SCAN
SCAN    JSR $FE89                derail DOS
        JSR $FE93
        LDY /UOD
        LDA #UOD
        JSR $DB3A
        LDA $CØ1Ø
.1Ø     LDA $CØØØ
        BPL .1Ø
        CMP #"U
        BEQ UP
        CMP #"u
        BEQ UP
DOWN    LDY #$23                      track
        LDX #Ø                       sector
        LDA #$FF                        inc
        BNE SETUP
UP      LDY #$FF                      track
        LDX #$F                      sector
        LDA #1                          inc
SETUP   STY TRACK
        STX SECT
        STA INC
        LDA #Ø
        STA VOL
        STA PTR
        LDA #1
        STA CMND
        STA FLAG tell getbyte to read sector
        JSR $FD8E
        JSR PTS
GO      LDX #Ø
        STX MIND
        LDA LEN
        STA LEN2
        LDA PTR
        STA MSTART
        LDA TRACK
        STA TØ
        LDA SECT
        STA SØ
.1Ø     JSR GETBYTE
        BCS .9Ø
        LDX MIND
        BNE .2Ø
        PHA
        EOR STRING
        STA EOR
        PLA
.2Ø     EOR STRING.X
        CMP EOR
        BNE .3Ø                    no match
        INC MIND
        DEC LEN2          matched this char
        BNE .1Ø
* ........................Complete match
        LDA TØ
        JSR PT
```

```
         LDA SØ
         JSR PS
         LDA #1Ø
         STA $24
         LDA MSTART
         JSR $FDDA
         LDA #":
         JSR $FDFØ
         LDA EOR
         JSR $FDDA
         JSR PTS
         LDA TRACK
         JSR PT
         LDA SECT
         JSR PS
* Go back & start at byte after beginning of
last match attempt, see if we need to go back a
sector.
.3Ø      LDY MSTART
         INY
         BEQ .4Ø    OK if getbyte will read in
                    new sector
         CPY PTR
         BCC .4Ø              OK if MSTART<PTR
         LDA PTR
         BEQ .4Ø     OK if next sector hasn't
                     been read
         LDA TØ
         STA TRACK
         LDA SØ
         STA SECT
         JSR RWTS
         LDY MSTART
         INY
.4Ø      STY PTR
         JMP GO
.9Ø      LDA $CØ1Ø
         JSR $FE84    I don't know why, but my
                      Apple goes bad if these
         JSR $FB2F    aren't both called
         JMP $A851               connect DOS
PTS      LDY /TS
         LDA #TS
         JSR $DB3A
         LDA $25
         STA VTAB
         RTS
PT       LDX #2
         .HS 2C
PS       LDX #7
         STX $24
         JMP $FDDA
GETBYTE  LDA $CØØØ
         CMP #$AØ
         BEQ GETBYTE                 pause
         CMP #$9B                Esc exits
         BEQ .2Ø
         LDY PTR
         BNE .Ø
         LDA MSTART            If MSTART=PTR=Ø,
                            already loaded sector
                            during last comparison
         ORA FLAG           load if 1st sector read
         BNE .1                        new sector
```

```
.Ø       LDA BUFFER,Y
         INC PTR
         CLC
         RTS
.1       LDA VTAB
         STA $25
         JSR $FC22                      Vtab
         LDY #Ø
         STY FLAG
         LDA SECT
         CLC
         ADC INC
         BPL .3
         LDA #$F
         INY                  flag new track
.3       CMP #$1Ø
         BNE .4
         LDA #Ø
         INY                       new track
.4       STA SECT
         TYA
         BEQ .5                 no new track
         LDA TRACK
         CLC
         ADC INC
         BMI .2Ø                        done
         CMP #$23
         BEQ .2Ø                        done
         STA TRACK
         JSR PT
.5       LDA SECT
         JSR PS
         JSR RWTS
         LDY #Ø
         BCC .Ø
         LDY /ERRMSG
         LDA #ERRMSG
         JSR $DB3A
         LDA ERR
         JSR $FDDA
         JSR PTS
         JMP .1
.1Ø      CLC
         RTS
.2Ø      SEC-
         done
         RTS
RWTS     LDA /IOB
         LDY #IOB
         JMP $3D9
UOD      .HS 8D
         .AS -/READ (U)P OR (D)OWN? /
         .HS ØØ
TS       .HS 8D
         .AS -/T$ S$/
         .HS ØØ
ERRMSG   .AS -/    ERR #$/ 3 spaces before err
         .HS ØØ
EOR      .BS 1byte which string is EORed with
             on disk
FLAG     .BS 1
INC      .BS 1  T/S increment: move up or down
LEN2     .BS 1             countdown to match
MIND     .BS 1                   match index
```

```
MSTART   .BS 1              match start index
PTR      .HS ØØ            pointer into buffer
TØ       .BS 1          track match started on
SØ       .BS 1
VTAB     .BS 1
*
IOB      .HS Ø16ØØ1
VOL      .HS ØØ
TRACK    .BS 1
SECT     .BS 1
         .DA DEVTAB
BUF      .DA BUFFER
         .HS ØØØØ
CMND     .BS 1
ERR      .HS ØØ
         .HS FE6ØØ1
DEVTAB   .HS ØØ01EFD8
BUFFER
*
         .OR $3ØØ
LEN      .BS 1
STRING
```

DISKSCAN hex dump

```
2222: 2Ø 89 FE 2Ø 93 FE                $6976
2228: AØ 23 A9 AB 2Ø 3A DB AD          $DF75
223Ø: 1Ø CØ AD ØØ CØ 1Ø FB C9          $914B
2238: D5 FØ ØC C9 F5 FØ Ø8 AØ          $7E77
224Ø: 23 A2 ØØ A9 FF DØ Ø6 AØ          $7638
2248: FF A2 ØF A9 Ø1 8C E3 23          $DD76
225Ø: 8E E4 23 8D D7 23 A9 ØØ          $74C6
2258: 8D E2 23 8D DB 23 A9 Ø1          $23E4
226Ø: 8D EB 23 8D D6 23 2Ø 8E          $3A7B
2268: FD 2Ø 1Ø 23 A2 ØØ 8E D9          $1C7D

227Ø: 23 AD ØØ Ø3 8D D8 23 AD          $5729
2278: DB 23 8D DA 23 AD E3 23          $59E8
228Ø: 8D DC 23 AD E4 23 8D DD          $23E2
2288: 23 2Ø 27 23 BØ 76 AE D9          $76C8
229Ø: 23 DØ Ø8 48 4D Ø1 Ø3 8D          $827F
2298: D5 23 68 5D Ø1 Ø3 CD D5          $684E
22AØ: 23 DØ 38 EE D9 23 CE D8          $41E4
22A8: 23 DØ DE AD DC 23 2Ø 1D          $ØØ4C
22BØ: 23 AD DD 23 2Ø 2Ø 23 A9          $852A
22B8: ØA 85 24 AD DA 23 2Ø DA          $14BA

22CØ: FD A9 BA 2Ø FØ FD AD D5          $4234
22C8: 23 2Ø DA FD 2Ø 1Ø 23 AD          $1D7C
22DØ: E3 23 2Ø 1D 23 AD E4 23          $45Ø4
22D8: 2Ø 2Ø 23 AC DA 23 C8 FØ          $CA22
22EØ: 1D CC DB 23 9Ø 18 AD DB          $93CB
22E8: 23 FØ 13 AD DC 23 8D E3          $A488
22FØ: 23 AD DD 23 8D E4 23 2Ø          $9EB5
22F8: A4 23 AC DA 23 C8 8C DB          $793B
23ØØ: 23 4C 6C 22 AD 1Ø CØ 2Ø          $DA78
23Ø8: 84 FE 2Ø 2F FB 4C 51 A8          $1ØF2

231Ø: AØ 23 A9 C2 2Ø 3A DB A5          $8399
2318: 25 8D DE 23 6Ø A2 Ø2 2C          $89EE
232Ø: A2 Ø7 86 24 4C DA FD AD          $C181
2328: ØØ CØ C9 AØ FØ F9 C9 9B          $27B6
233Ø: FØ 7Ø AC DB 23 DØ Ø8 AD          $Ø218
2338: DA 23 ØD D6 23 DØ Ø8 B9          $5CC4
234Ø: F4 23 EE DB 23 18 6Ø AD          $CF6F
```

```
2348: DE 23 85 25 20 22 FC A0        $BEE6
2350: 00 8C D6 23 AD E4 23 18        $6B7B
2358: 6D D7 23 10 03 A9 0F C8        $063C
2360: C9 10 D0 03 A9 00 C8 8D        $A8C3
2368: E4 23 98 F0 13 AD E3 23        $43E3
2370: 18 6D D7 23 30 2C C9 23        $1DDF
2378: F0 28 8D E3 23 20 1D 23        $7C1E
2380: AD E4 23 20 20 23 20 A4        $E01F
2388: 23 A0 00 90 B2 A0 23 A9        $E2E9
2390: CB 20 3A DB AD EC 23 20        $744B
2398: DA FD 20 10 23 4C 47 23        $EB6A
23A0: 18 60 38 60 A9 23 A0 DF        $E58F
23A8: 4C D9 03 8D D2 C5 C1 C4        $A6B3
23B0: A0 A8 D5 A9 D0 A0 CF D2        $206C
23B8: A0 A8 C4 A9 CF D7 CE BF        $1519
23C0: A0 00 8D D4 A4 A0 A0 A0        $CF02
23C8: D3 A4 00 A0 A0 A0 C5 D2        $CC2D
23D0: D2 A0 A3 A4 00 00 00 00        $F2C0
23D8: 00 00 00 00 00 00 00 01        $53C0
23E0: 60 01 00 00 00 F0 23 F4        $A1A3
23E8: 23 00 00 00 00 FE 60 01        $B9E1
23F0: 00 01 EF D8                    $B12C
```

Jose A. Montano

Raves for the MacIntosh Extended Keyboard

I have just purchased Apple's new MacIntosh extended keyboard and attached it to my GS to see what would happen. Lo and behold! Nothing happened (nothing wrong, that is). It works great on the GS. The key action is much smoother and you have 21 extra keys to assign whatever functions you want. In addition, the CONTROL, OPTION and APPLE keys are duplicated on the lower right hand side where the GS arrow keys are. The arrow keys are to the right of these (in a more sensible layout, if I do say so myself).

For any of your readers who are interested in using the new keyboard, here are a few things I have found out about it.

Extra Keys and Their ASCII Values

F1	z	Chr$(122)	F12	o	Chr$(111)
F2	x	Chr$(120)	F13	i	Chr$(105)
F3	c	Chr$(99)	F14	k	Chr$(107)
F4	v	Chr$(118)	F15	q	Chr$(113)
F5	`	Chr$(96)	HELP	r	Chr$(114)
F6	a	Chr$(97)	Del	u	Chr$(117)
F7	b	Chr$(98)	HOME	s	Chr$(115)
F8	d	Chr$(100)	END	w	Chr$(119)
F9	e	Chr$(101)	PAGE UP	t	Chr$(116)
F10	m	Chr$(109)	PAGE DOWN	y	Chr$(121)
F11	g	Chr$(103)			

All extra keys set Bit 4 in the KEYMOD Register at location $C025 (49189) indicating they were pressed on the numeric keypad.

For most BASIC applications there is no difference between F1 and "z", for example. To see the difference you must query the KEYMOD Register with suitable assembler instructions such as:

```
LOOP LDA $C025 Get the current key status
     BIT #$10  Was key pressed on keypad?
     BEQ LOOP  If not, recheck key status
     LDA $C000 If so, load key into
               Accumulator
     STA $C010 Clear the keyboard strobe
     ...       continue with your code
```

One possible Applesoft BASIC differentiation would be:

```
100 GET A$
110 KM = PEEK(49189)
120 IF KM <> 16 THEN PRINT CHR$(13): GOTO 100
130 PRINT A$
...continue with your code....
```

For the sake of interest, the following chart shows how the KEYMOD register is bit-mapped.

```
BIT VAL DESCRIPTION
7   1   If  was pressed
6   1   If Option key was pressed ()
5   1   If the modifier keylatch has data
        (any key pressed)
4   1   If a key on the numeric keypad has
        been pressed
3   1   If a key has been held down long
        enough to engage repeat
2   1   If the Caps Lock key is down
1   1   If the Shift key is down
0   0   Unknown (or Unused)
```

This keyboard could add whole new dimensions to commercial applications if software designers for the IIgs would program with the new keyboard in mind. You could also enhance your own personal masterpieces by using the function keys for program flow instead of the cryptic keypresses that must be used now. The new keyboard also allows for function key overlays to let you label each function key the way you want to.

Apple's new extended keyboard is a boon not only for MacIntosh users, but for Apple IIgs users, as well, and the sooner software companies realize it the better.

Jolly Roger of The Curse

Softkey for...

Microzine #14 Fun House

Scholastic

■ Requirements

□ Sector editor

When asked to deprotect this software I first consulted old issues of COMPUTIST and found it was not listed. It was listed in the *Copy II Plus* parameter list, but a copy made with this program (or any bit copier) wouldn't work. Having exhausted the "easy" methods, it was time to do an original crack.

Too many softkeys give only the step-by-step procedures and not how the crack was done. Fear not - details follow!

The first step is to determine the type of protection. Was an altered DOS or a nibble count used or were other nefarious and sneaky tricks employed?

Upon boot, an Applesoft prompt appeared which would mean DOS is reasonably intact. The disk could be cataloged under normal DOS 3.3 - normal DOS may be used. The disk could be copied with COPYA, but the copy continually reboots. This leads us to a nibble count or some sort of disk signature checking routine as the major protection.

Copy II Plus revealed that the boot program was named HELLO. This was beginning to look too easy. Hmmmm, when DOS 3.3 is booted and then the HELLO program loaded the only thing that appeared with the LIST command is 0 REM.

After a CALL-151 to enter the monitor, a check of $0067 shows that the listing should begin at $0801 - all normal. However, looking at the code beginning at $0801 shows that the authors have screwed around with the BASIC line pointers to prevent us from LISTing the HELLO program and thus discovering the file that is called which contains the CALL to the protection code.

Well, it would have been a simple, but perhaps tedious, task to un-diddle the line pointers but an easier way around this was possible. I simply invoked MONICO and then ran the HELLO program. The first file loaded was W.SPC, a suspiciously short binary file. I went back to *Copy II Plus* and mapped the disk to find the location of the offending W.SPC file. It was located on track $05, sectors $02 and $01. Another way to do this would have been to go to track $11 (catalog) and read the location from there, but I'm lazy.

Anyway, using the sector editor, I went to track $05, sector $02 and found the Track/Sector list for the file which just said that the file was located on sector $01. Since this was a binary file, the first two bytes at the beginning of track $01 were the start address of the file in memory and the next two were the file length. The actual code began at byte 5 aand was a JSR to wherever. I didn't even look. The code was a series of stack pushes and pulls - pretty obvious this was protection code. I just slapped an 18 60 into bytes 5 and 6 and booted the disk. It worked fine. They didn't even bother to checksum the protection code to see if it was changed. Of course, you could un-diddle the HELLO program and remove the BASIC CALL to the start address of the W.SPC file but that's too much work.

Step-By-Step

1 Copy the disk with COPYA or any disk copy program.

2 Make the following sector edit:

Trk	Sct	Byte(s)	From	To
$Ø5	$Ø1	$Ø5-Ø6	2Ø XX	18 6Ø

John Wulfken

Softkey for...

Hardball IIgs

Accolade

Requirements

- □ Apple IIgs with 512K
- □ A blank 5¼" disk
- □ A sector editor like *Copy II Plus*

I found a simple approach to deprotecting *HardBall GS* by first removing all references to the reset vector, and loading in all values changed by the protection. The protection itself starts at hex address $2CØØ with two calls to hex address $6B5D. If you would like to trace the protection, I recommend that you purchase a copy of Apple IIgs Firmware Reference the official publication. Chapter 7 contains a wealth of information on deprotecting GS software.

The code at $6B5D is a nice piece of work by Accolade. It loads in as a JSR FFFF and is changed during program execution to a JSR C5ØD. This is a smartport call much like the ProDOS call JSR BFØØ call.

Step-By-Step

1 Make a copy of your original disk. Then format a 5¼" disk. Move the file HARDBALL/SYSTEM/START from your copy disk to your 5¼" disk.

2 Using a sector editor search for 8F F3 Ø3 ØØ and replace with EA EA EA EA.

3 Copy this file back to your disk and reformat your 5¼" disk. Copy the file HARDBALL/HARD.CODE to your 5¼" disk and search for 9C F3 Ø3 9C F4 Ø3 and replace with EA EA EA EA EA EA. You will now be able to **⌘RESET** and the disk will not reboot.

4 Search for Ø8 8B ØB C2 2Ø A9 5D 6A 5B E2 2Ø AD ? and replace with Ø8 28 A9 35 8D 78 Ø3 A9 1E 8D 79 Ø3 A9 DØ 8D 3B ØØ A9 12 8D FØ ØØ A9 Ø1 8D F4 ØØ A9 BØ 8D F6 ØØ A9 ØD 8D F7 ØØ A9 22 8D F8 ØØ 8D EC ØØ A9 19 8D EA ØØ A9 ØB 8D EB ØØ A9 A8 8D ED ØØ A9 Ø3 8D Ø2 95 A9 F8 8D Ø4 95 A9 48 8D Ø5 95 A9 41 8D Ø6 95 8D ØA 95 A9 52 8D Ø7 95 A9 44 8D Ø8 95 A9 42 8D Ø9 95 A9 4C 8D ØB 958D ØC 95 A9 ØØ 8D F5 ØØ 8D F1 ØØ 8D EE ØØ 8D ØØ 95 8D Ø1 95 8D Ø3 95 8D ØD 95 8D ØE 95 A2 E3 AØ 1F 6Ø EA EA EA

5 Now copy the file back to your 3½" disk and that is it.

Winter Games GS also uses the same type of protection.

Doug Parrish

Softkey for...

Typewriter

Channelmark

Requirements

- □ *Typewriter* original
- □ 1 blank disk side
- □ Sector editor (*Copy II Plus* is fine)

Jim Hart, COMPUTIST #52, page 23, did the work on *Grid Designer* and I applied it to another one of Channelmark's disks.

A friend badly needed *Typewriter* backed up, so he gave it to me. Channelmark had always caused me trouble, but Jim Hart's crack for *Grid Designer* suddenly gave me new hope.

Before I go further, I'd like to pass on a tip I have picked up while working with deprotecting schemes. There is definitely a difference between a "blank" disk and one you're going to reuse. Blanks have no leftover code from a previous program, while "used" disks do. Although I don't know the ins and outs of disk code deleting that well, I suspect that the code left on a disk can cause deliberate misreads when the copy procedure is trying to lay down accurate timing and epilog bytes.

Unless special hardware was used to record an extra strong pattern on a disk, initializing will erase whatever was on a disk by writing new patterns. However, deleting all the files on a disk does not remove the data from that disk. It only deletes part of the pointer to that data so that the catalog command will show no files. If you need a blank, unformatted disk, use a bulk tape eraser...........RDEXed

1 Use *Copy II Plus* to copy the program side of *Typewriter* to the blank disk.

2 Using the sector editor, make the following two changes to the code on the copy:

Trk	Sct	Byte(s)	From	To
$11	$ØE	$3E-4Ø	AD E9 CØ	18 9Ø 4C
$1E	$ØC	$3E-4Ø	AD E9 CØ	18 9Ø 4C

One other Item

(?) On Three puts out a program called *Graphics Manager*. It reduces and enlarges clip art and allows it to be placed anywhere on a page, or so the flyer says. Over the past year and a half, I have tried to get the program to work with my setup: Apple IIe (unenhanced), RamWorks II with 1 meg, 2 Disk II's and a parallel Okidata 92 printer. I even photocopied the parallel Grappler Plus manual and sent it to them in hopes they could rewrite the code. They tried. So far, nothing. I finally gave up and I guess they did too after the third or fourth disk got here.

Can anybody explain to me why this won't work, what I have to do, what code I could change? They say it's the parallel interface. But the Grappler Plus works just fine on everything else, except their disk. Do I have to reset the DIP switches under the printer's housing? Has anyone out there solved this one yet?

Keep up the fantastic format! It's superb. COMPUTIST has helped me so many times, it has more than paid for the subscription price.

Keith Parker

Softkey for...

Bard's Tale II

Electronic Arts

Requirements

- □ A blank disk
- □ A sector editor
- □ Copy program that will ignore errors (*Locksmith's Fast Backup*)
- □ *Bard's Tale II: Destiny Knight* original

When I heard about the release of *Bard's Tale II*, I couldn't believe it would have better graphics than *Bard's Tale*. I told all of my friends, "Different graphics maybe, but better ones? No way!". I was definitely wrong! *Bard's Tale II* does have better graphics. *Bard's Tale II* also has new magic spells, more towns, dungeons, and monsters.

The protection used on this game is very similar to the protection used on *Boulderdash*. It seems that Electronic Arts is starting to repeat their protection schemes.

To start off, I copied the disk with *Locksmith's Fast Backup*. This told me that track $Ø6 was the error track. I then booted up my copy to see how far it would get. After the EA logo came up the program did it's first check. I jumped into the monitor using *Senior Prom* and looked at $AØØØ. I wrote down the first 3 bytes and then searched my disk for the sequence. I found it on track $Ø1, sector $ØF starting at byte $ØØ. I changed the jump code by putting a 6Ø in place of the 4C. Now I had to correct the checksum by changing byte $FF of track $1Ø, sector $Ø7 from ØF to Ø2.

I then booted up my disk. The program started to load exactly like the original, until it tried to load in the player screen (the main part of the game). The grinding noise, that came from my drive, was enough to give anybody a heart attack. I jumped into the monitor using *Senior Prom* to see what was going on. I looked at the softkey for *Boulderdash* again. In that softkey, the protection code started with the byte sequence A9 14 48. I used the Search Memory function with the *Senior Prom* to scan for the sequence. No luck. So I decided to scan for the byte sequence A9 = 48 (The "=" is a wildcard, meaning any byte can be there.). The *Senior Prom* found the sequence in four locations. Since one of the locations was in my computers ROM, that eliminated that choice. I listed the other three locations and found that two of them were identical. EA is known for making such routines. Here is what the routine looks like:

```
2E44  A9 ØØ     LDA #$ØØ
2E46  48        PHA
2E47  A5 44     LDA $44
2E49  D9 A4 B3  CMP $B3A4,Y
2E4C  9Ø 12     BCC $2E6Ø
2E4E  F9 A4 B3  SBC $B3A4,Y
2E51  85 44     STA $44
2E53  A5 45     LDA $45
2E55  E9 ØØ     SBC #$ØØ
2E57  85 45     STA $45
2E59  68        PLA
2E5A  69 ØØ     ADC #$ØØ
2E5C  48        PHA
2E5D  4C 47 AE  JMP $AE47
2E6Ø  68        PLA
2E61  Ø9 BØ     ORA #$BØ
2E63  2Ø ED FD  JSR $FDED
2E66  88        DEY
2E67  1Ø DB     BPL $2E44
2E69  6Ø        RTS
```

The same routine was found at location $3144. I then pulled out my sector editor and searched for the byte sequence A9 ØØ 48. I found it at track $ØF, sector $Ø1 and track $1Ø, sector $ØE. I replaced the jump with 18 6Ø 4Ø and then booted up the copy. It worked like a charm!

1 Copy the disk with a copier that will ignore errors.

2 Boot up your sector editor and make the following changes (Be sure to write the sectors back to disk.):

Trk	Sct	Byte(s)	From	To
$Ø1	$Ø7	$FF	ØF	Ø2
	$ØF	$ØØ	4C	6Ø
$ØF	$Ø1	$44	A9 ØØ 48	18 6Ø 4Ø
$1Ø	$ØE	$44	A9 ØØ 48	18 6Ø 4Ø

Go out and destroy the evil Archmage Lagoth Zanta.

Greg Poulos

How To Modify All Infocom Games For Lower-case

Many years ago, there was a file floating around that explained how to modify Infocom games to produce lower case characters during game play. The procedure outlined in that file worked beautifully on Infocom's first releases.

But, alas, times changed and Infocom moved things around on the disk. In the newer versions on their games, you are able to select 4Ø or 8Ø column output. Anyone who has tried the well-known old techniques to get lower-case has probably come up with zero success.

Well, I did some searching around and found that although the location of the code may have jumped around a bit, the code that changes the output is the exact same on all Infocom games.

The reason Infocom makes all the 4Ø column output upper-case, is because they think there are still people that have no lower-case chip in their Apple. Somebody ought to tell them that 99% of Apple owners have lower-case.

There are three chunks of code that must be altered to create a nice looking lower-case output. The first routine prints the ever-familiar inverse bar at the top of the screen. This contains your location and either your score/number of moves or the time. This routine must be altered to not print in inverse, for as we all know, inverse lower-case characters don't agree with the Apple, although in 8Ø columns you get inverse/lower-case. This is the only thing you'll lose if you decide to modify your Infocom games to lower-case, there'll be no pretty inverse bar, not even in 8Ø columns.

The next routine is a big one. This is the routine that actually converts all the characters to upper case. This is eleven bytes long (decimal) and will have to be completely NOPed out with eleven EA's.

The final routine determines the size of the window at the top of the screen for the text to scroll under. This is currently one line (the inverse bar). For easier readability, you'll want to change this to two lines, so there's a blank line in between the location description, score, etc., and the actual game text itself.

So with all the explanation out of the way, let's go to work. You must first have a cracked copy of the game. Nowadays, Infocom only protects their two-sided, 128K games like *Trinity*, but in case you have an old *Zork* or something, deprotect it first. (See Book of Softkeys Vol. 1 for *Zork*).

Run a sector editor like the one in *Copy II Plus* that has search capability. Insert your Infocom disk.

The Inverse Routine

Search for hex bytes A9 3F 85 32 A9 1Ø. Once you find these (somewhere on the first two or three tracks, probably), change the 3F to FF.

The Window Routine

Search for hex bytes A9 Ø1 85 22 85 EØ. Once you find these, change the Ø1 to an Ø2. Note: Ø2 is recommended. You may change this to a higher number to get a wider gap. NOTE: If you cannot find the "Window" routine, search for A9 Ø1 85 22 instead.

The Upper Case Routine

Search for hex bytes C9 6Ø 9Ø ØB C9 8Ø BØ Ø7. Once these are found, put the cursor on the first byte (the first C9) and enter eleven EA's.

That's all there is to it. Something you might try doing is allowing the inverse bar in 8Ø columns, but making it normal in 4Ø columns. I haven't been able to do this as of yet.

David Burns and Brian Sparks

Softkey for...

California Games
Championship Wrestling
The Movie Monster Game

Epyx

■ **Requirements**

□ A blank DOS 3.3 formatted disk (double sided)
□ Fast copier (that ignores errors)
□ A sector editor (I used Copy II plus)

California Games is a really rad game from Epyx that enhances the Summer games line. After about an hour of playing, you will start using words like gnarly, tubular, and awesome as part of your everyday vocabulary. Just follow these instructions for a deprotected copy:

1 Fast copy both sides of *California Games* with any fast copier that ignores errors.

2 Make the following sector edits:

Trk	Sct	Byte(s)	From	To
$ØØ	$Ø5	$FØ-F7	ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ	FC EE EE FC E7 EE FC E7
		$5E-6Ø	BD 8C CØ	AD F1 BB
		$6E-7Ø	BD 8C CØ	B9 FØ BB

3 Write these changes back to disk.

Side 2 is not protected. This softkey also works with two other Epyx games I recently bought, *Championship Wrestling* and *The Movie Monster Game*.

Robert Wilson

Softkey for...

Deep Space

Sir-Tech

■ **Requirements**

- □ Apple II series with 64K
- □ *COPYA*
- □ Sector editor
- □ Blank disk

Deep Space is a space-combat simulator that features smooth, convincing 3-D graphics as you fly four different missions in an area of the astroid belt that lies between the orbits of Mars and Jupiter.

The disk is virtually unprotected and can be copied with any copier that will ignore the read error on track $22. The resulting copy will not work however, even though track $22 contains no useful code. To make a long search short, a little detective work led me to the entry point for the protection routine. I found it on track $ØC, sector $Ø9.

Step by step

1 Boot your DOS 3.3 system disk.

2 Tell DOS to ignore checksum and epilog errors and use *COPYA* to copy the disk.

```
POKE 47426,24
RUN COPYA
```

3 Make the following sector edits to the copy you just made.

Trk	Sct	Byte(s)	From	To
$ØC	$Ø9	$8B-8C	A9 ØD	18 6Ø

That's it. The copy should now work and is *COPYA*-able.

A.P.T. for...

Deep Space

Sir-Tech

Now here are some A.P.T.s that may make your new copy of *Deep Space* a bit more interesting.

Unlimited fuel:

Trk	Sct	Byte(s)	From	To
$1B	$ØE	$E2-E3	C6 4F	EA EA

Unlimited hyperspace drive:

Trk	Sct	Byte(s)	From	To
$21	$Ø8	$E7-E8	E5 Ø3	38 EA

Unlimited missiles:

Trk	Sct	Byte(s)	From	To
$1C	$Ø2	$66-67	FØ 28	EA EA
		$BD-BF	CE 8F 6Ø	EA EA EA

To stop those irritating attacks on your bases while flying the Outpost Mission:

Trk	Sct	Byte(s)	From	To
$21	$Ø4	$8C-8D	C6 55	EA EA

An indestructable shield is available by changing the following:

Trk	Sct	Byte(s)	From	To
$1C	$Ø1	$EE-FF	E5 1Ø	EA EA

Also, a dot of transparent tape punched out with a hole punch while the tape is still stretched out on the despenser, then placed at the right spot on the screen, makes an excellent gun sight for the missiles and laser.

A co-author of *Deep Space*, has an excellent new flight simulator in the stores. It is being sold as *Chuck Yeager's Advanced Flight Trainer*, this one can teach you to master other flight simulators, and allows you to fly 14 different airplanes ranging from the Sopwith Camel F-1 to experimental rocket planes that can take you to the edge of space.

While *Advanced Flight Trainer* is published by Electronic Arts, it contains none of their well published locks. Headers that change with every sector, and a few other goodies too, it looks like a challenge for the advanced hacker.

How about it, ladies and gentlemen, is there anyone out there who can crack this one?

Zorro

Softkey for...

Arcticfox

Electronic Arts

A friend of mine asked me to try and crack his original of *Arcticfox* before it goes "kaputt". I tried several of the softkeys from COMPUTIST, but none were able to crack it.

Knowing that Electronic Arts uses their nibble check routine on track $Ø1, and that this basic type of protection is used on most EA games, I looked at the softkey for *The Bard's Tale* in COMPUTIST #51 to find some similarities.

As I expected, some of the bytes were the same as in the *Bard's Tale* softkey, so I changed them to the values following them and it worked. Here's how it goes:

Trk	Sct	Byte(s)	From	To
$Ø1	$ØB	$47-49	2Ø ?? ??	18 6Ø 42
	$ØE	$47-49	2Ø F8 AØ	18 6Ø 4Ø
		$4C-4E	2Ø Ø3 AØ	18 6Ø 4B
	$ØF	$ØØ-Ø2	4C 69 AØ	18 6Ø DD
		$6F-71	4C 69 AØ	18 6Ø DD

Playing Tips for...

Arcticfox

Here is something that may be useful to new ArcticFox gamesters, unless you know about this already. Ever notice the two little boxes with the numbers in them on the lower-right of the screen? Those are for Latitude and Longitude. The main fortress lies around the following numbers (in order, top to bottom on the screen):

```
Ø53  Ø6
83   Ø1
```

From there you should be able to see the main fortress, and destroy it with a missile. Be careful, the alien base is highly fortified all around. Also DO NOT get too close to the fortress or else it will deploy homing mines (similar to floating mines, but bigger and more ominous) that will come out in waves and destroy you. The fortress has a limitless supply of these, so be wary.

Jim S. Hart

Softkey for...

BoulderDash Construction Set

Epyx

■ **Requirements**

- □ *Boulder Dash Construction Set* (BDCS)
- □ An initialized DOS 3.3 disk
- □ COPYA
- □ Disk searcher/sector editor (I used *Copy II Plus*)
- □ Nibble editor
- □ Disk scanner (like *Locksmith*'s Quick Disk Scan, used for determining format of disk)
- □ Optional Recommended reading: Beneath Apple DOS (BAD) by Don Worth and Pieter Lechner

This is a softkey which I have extended a bit to help out those of you who are just beginning the art of deprotection. I've read the complaints readers have about softkeys either being too short or not explaining what has been changed. I will go through several steps that I usually follow which will culminate in a softkey.

Before we start off, let's get something straight. If terms like prolog, epilog, disk formatting, checksums, and direct disk accessing are foreign to you, then I suggest you go out and pick up a copy of Beneath Apple DOS (BAD). It really is an excellent book on the subject of disk input/output and covers the structure of DOS 3.3 quite well. After you have bought your copy, there are three things to do: read, read, and read more. Don't expect the material to make sense at first. It takes a lot of work and studying before 'the light comes on'. If it's worth learning, it's worth working at.

Another thing that helps tremendously is back issues of COMPUTIST. Sure, a lot of the softkeys are short, but many are not and several go in depth into the protection schemes. In addition, one of those softkeys may be for the same type of protection scheme that is on your disk! Learn as much as you can about protection schemes in general. This is the process I went through when I started out in the deprotection business, and today I think I am quite proficient at it. It just takes diligence and hard work.

The Softkey Process

The first thing I do is to find out what kind of protection the disk has. There are two basic types of protection schemes: signature checking and format alterations.

Signature Checking

A disk protected with a signature check can usually be copied with any whole disk copying program, such as the Locksmith fast copier or *COPYA*. You are able to copy the disk, but for some reason the copied disk will not work.

What happens is that when the original disk was written, special disk drive hardware was used to write a specific byte or bytes to the disk. The Apple disk drive would be able to read these bytes but not write them.

Types of signature checking include the infamous nibble count, bit insertion, and phantom bits. All are *COPYA*-able (to a large extent) but differ in the way they check for things. Again, special drive hardware is used to put data on the disk in such a way that a regular disk copy will leave out the bytes the protection routine is looking for.

A nibble count looks on the disk for a certain byte sequence and then starts counting bytes until another special byte sequence is found. If the number of bytes between the two sequences is out of a certain tolerance range (determined by the protection scheme), then the protection scheme 'knows' that the disk is a copy and takes appropriate measures. Usually this results in a reboot, but other things can happen too. For example, say you need to get the magic scroll in order to get through a certain room, and if you do not have the scroll then you can not advance in the game. The protection scheme 'finds out' that the disk is a copy and alters some code so that you can never find the magic scroll, ergo you can never get past that certain room and you are stuck, unable to go any further. Keep this in mind when you supposedly deprotect a game: make sure you have removed the protection entirely.

Bit insertion and phantom bits (sometimes referred to as weak bits) are beyond the scope of this article. I suggest you bone up on chapter 3 in BAD and go through lots of old COMPUTISTs for help on these.

Format Alterations

Format alterations are some of the easier (to me) types of protections to deal with, if they are the only protection a disk has. A format alteration is one in which the actual format of the disk has been changed from the normal. One of the following generally has been changed on these disks: address prolog, data prolog, address epilog, data epilog, address checksum, data checksum, or the nybble translate tables.

There are some disks, like the *Essential Data Duplicator* and the *Flight Simulator II*, that have drastic format changes which may or may not include the above items. These nasty disks deviate away from standard 16 sector format and I will not go over how to deprotect them since each one must be individually examined.

Changes that have been made to either the prologs, epilogs, or the checksums are found out easily with the help of a nibble editor. A nibble editor disregards the format of the disk and just reads in an entire track at a time and then displays it in raw nibble format.

Looking at the display, you should find the following: a large group of bytes in normal text, 15-2Ø bytes in inverse text (usually with the value of $FF), 13-16 bytes in normal text, and 5-1Ø bytes of inverse text. This pattern should repeat itself 16 times on the track since there are 16 sectors per track.

The large group of normal bytes is the data area. The data area's first three bytes compose the data prolog. The third byte from the end is the checksum. The last two bytes compose the data epilog. The 15-2Ø bytes of inverse text are the sync gap between the data and address areas.

The next group of 13-16 bytes are the address header which is broken up into address header (3 bytes), volume # (2 bytes), track # (2 bytes), sector # (2 bytes), checksum (2 bytes), and the address epilog (2 bytes). The value of the checksums, non-prolog, and non-epilog data depend on the information contained within the sector.

The values of the prologs and epilogs on a normal DOS 3.3 disk are:

```
Address prolog....................D5 AA 96
Address epilog......................DE AA
Data prolog.......................D5 AA AD
Data epilog.........................DE AA
```

Go ahead and try a nibble editor out on a normally formatted DOS 3.3 or ProDOS disk and hunt for the above bytes. Get proficient at finding them and when it comes time to find them on a protected disk, you will generally not have a problem.

Another tool to use is a disk scanner, such as *Locksmith*'s Quick Disk Scan. What this shows you are the sync gaps and sectors on a track. The sync gaps are represented by white dots and the sectors are represented by no dots. Looking at the output, you can count the number of sectors and if it is around 16-17 then you have a 16 sector format disk. Again, try the disk scanner on a normal DOS 3.3 or ProDOS disk to see what a 16 sector disk's display looks like. DOS 3.2 disks have larger sector sizes and less sync gaps and with some experience it becomes easy to distinguish between 13 and 16 sector format.

Other types of format alterations include track synchronization and non standard tracking (half tracks, quarter tracks). Half tracking is used by Electronic Arts quite frequently. Tracks $5, $5.5, and $6 are all formatted the same way and the Apple drives can read half tracks with (usually) no problem. The bug is when it comes to writing half tracks, the drive does not have the reliable precision to write a half track without writing over the data a half track away on both sides.

Track sync involves writing data to a disk in such a way that when the protection scheme moves the drive head from one track to another, it expects to find the correct data immediately. I will not go over format alterations that involve anything other than 16 sector disks because once again, each disk must be handled individually.

Disks protected with alterations to their prologs or epilogs are generally easier to deprotect. Often, all you have to do is read the disk using their format and then write to another disk using normal 16 sector format. *Super IOB* is ideal for this and I refer you to Ray Darrah's article in COMPUTIST #32 for directions on what it does and how to write a controller for it. After the disk has been converted to normal format, all that is required is to either put a normal DOS onto it or edit the disk's read routines so that it can now read in the normal format. DOS 3.3's disk read routines are located around $B8ØØ-$B95Ø or so. Boot DOS 3.3 and look at them sometime using the monitor list (L) command.

The routine looks something like this:

```
B8E1- BD 8C CØ  LDA $CØ8C,X
B8E4- 1Ø FB     BPL $B8E1
```

What these 5 bytes of code do is to read a byte off of the disk. Usually, right after these bytes is a CMP statement that compares what was read to a value. If that value is anything other than the normal prolog or epilog bytes, it is immediately suspicious.

For example, say you deprotected a disk with a data prolog of D5 AA CD. You've already converted the disk to normal 16 sector format. Now, you have to find where it was reading in that CD value. Using your disk searcher/sector editor you searched through the disk, starting at track $ØØ sector $ØØ, for the byte sequence 8C CØ. You found it and a disassembly of the code around it looked something like this:

```
E8Ø- BD 8C CØ  LDA $CØ8C,X
E83- 1Ø F8     BPL $E8Ø
E85- C9 CD     CMP #$CD
E87- DØ Ø3     BNE $E8C
```

Well, there is the code that read in the CD byte. All you have to do is to change the C9 CD to C9 AD and the disk should now have the ability to read itself.

NOTE: disks protected with format alterations may also have signature checks too, so don't forget to thoroughly check your deprotected disk.

Now, to the Main Attraction

Now that we have gone over several different types of protections, it is time to deal with the title program: BDCS. Interestingly enough, BDCS is protected with the EXACT same protection scheme as *California Games*. In fact, I tried my *California Games* deprotection in COMPUTIST #52 and it deprotected BDCS perfectly!

Step One

First of all, the disk is protected with a format alteration. *COPYA* doesn't work and the *Locksmith* fast copier comes up with lots of inverse characters (not a good copy). So, it's time to load up the trusty disk scanner. This reveals a 16 sector format. A quick check with the nibble editor reveals that the alteration is not a prolog alteration. Back in my *Information Master* softkey (COMPUTIST #43, page 23), I listed what bytes to change to ignore certain bytes and errors. **POKE 47426,24** (or B942:18 from the monitor) disables DOS 3.3's epilog and checksum error checking. This is what we need. After typing in the POKE, I then typed **RUN COPYA** and copied the BDCS disk onto a blank. Try to boot the disk up and you are rewarded with a reboot a few seconds after the disk boots. Ah hah! Another protection!

Looking for the Other Protection

When looking for a signature check, you can either boot code trace the disk (if the check is early in the boot) or search the disk for the byte sequence 8C CØ (direct disk access code). Since the reboot occurs fairly early in the boot process, I decided to boot code trace the disk, but not in the normal fashion. I used my sector editor to read in track $ØØ sector $ØØ and then disassembled the code in the sector starting with relative byte $Ø1. Refer to pages 5-6 to 5-8 and 8-1 to 8-3 in BAD for more information on the booting process. Looking down the disassembly, I was only interested in JMP commands. At relative byte $4A, there is a JuMP to $BBØØ. Now, if you look at track $ØØ sector $ØØ of a normal DOS 3.3 disk, there should be an indirect JuMP to $8FD (which finally ends up JuMPing to $B7ØØ) at relative byte 4A. In fact, up to that point the BDCS and normal boots are the same. Hmmm. Could it be that a signature check of some sorts is at $BBØØ, and when it is done it will JuMP to $B7ØØ? Well, let's take a look. On the disk, $BBØØ corresponds to track $ØØ sector $Ø5. Read in that sector using your sector editor and you will see the following disassembly. I have commented the code a bit in the hopes that you might better understand the protection.

Main Protection Code ($BBØØ)

```
ØØ LDA #$ØØ     Zero locations $FØ-F7.
Ø2 LDX #$FØ
Ø4 TXS
Ø5 STA $ØØ,X
Ø7 INX
Ø8 BNE $BBØ5
ØA LDA #$ØA     set up # of retries
ØC STA $FC      retries location
ØE LDX $2B      get slot #
1Ø LDA $CØ89,X turn drive on
13 LDA $CØ8E,X enable READ mode
16 LDA #$8Ø
18 STA $FD
1A DEC $FD
1C BEQ $BB98    Bad
1E JSR $BBF5    find correct track
21 BCS $BB98    Bad
23 LDA $F9
25 CMP #$ØA
27 BNE $BB1A
```

Check the disk for a certain sequence of header bytes.

```
29 LDY #$ØØ
2B LDA $CØ8C,X
2E BPL $BB2B
3Ø DEY
31 BEQ $BB98    Bad
33 AND #$ØØ
35 BNE $BB2B
37 LDA #$ØØ
39 LDA $CØ8C,X
3C BPL $BB39
3E DEY
3F BEQ $BB98    Bad
41 AND #$ØØ
43 BNE $BB39
45 LDA $CØ8C,X
48 BPL $BB45
4A AND #$ØØ
4C BNE $BB98    Bad
4E LDA $CØ8C,X
51 BPL $BB4E
53 AND #$ØØ
55 BNE $BB98    Bad
57 LDA $CØ8D,X
5A LDY #$1Ø
5C BIT $8Ø
5E LDA $CØ8C,X
61 BPL $BB5E
63 DEY
64 BEQ $BB98    Bad
66 AND #$ØØ
68 BNE $BB5E
6A NOP
6B NOP
6C LDY #$Ø7     Start at $F7, go down to $FØ
6E LDA $CØ8C,X
71 BPL $BB6E    Correct sequence of header
73 STA $ØØFØ,Y bytes found, so load $FØ-$F7
76 NOP          with sequence of bytes found
77 DEY          after header byte sequence.
78 BPL $BB6E
```

Decode next stage ($B7ØØ-B9FF) & execute.

```
7A LDX #$Ø3
7C LDA #$ØØ
7E TAY
7F STA $F8
81 LDA #$B7     Page to start decoding.
83 STA $F9
85 LDA $FØ,X    Get value from $FØ-$F3.
87 EOR ($F8),Y EOR byte
89 STA ($F8),Y Put decoded byte back.
8B DEY
8C BNE $BB85
8E INC $F9      New page to decode.
9Ø DEX
91 BPL $BB85
93 LDX $2B      Get slot#.
95 JMP $B7ØØ    Jump to decoded next stage.
```

Come here if something bad was detected.

```
98 DEC $FC      decrement # of tries loc.
9A BEQ $BB9F    have we used up our tries?
9C JMP $BB16    no, try again.
9F INC $Ø3F4    yes, alter RESET vector to
                reboot.
A2 JMP $BB95    next stage not decoded, so
                crash.
```

Seek correct track subroutine.

```
A5 LDY #$FD     # of tries to find D5 AA 96.
A7 STY $FØ      store it.
A9 INY          increment # of tries.
AA BNE $BBBØ    more tries?
AC INC $FØ      no, increment bad result
                location.
AE BEQ $BBED    correct track not found.
```

Look for D5 AA 96 sequence.

```
BØ LDA $CØ8C,X
B3 BPL $BBBØ
B5 CMP #$D5
B7 BNE $BBA9
B9 NOP
```

```
BA LDA $CØ8C,X
BD BPL $BBBA
BF CMP #$AA
C1 BNE $BBB5
C3 LDY #$Ø3
C5 LDA $CØ8C,X
C8 BPL $BBC5
CA CMP #$96
CC BNE $BBB5
```

Found correct track.

```
CE LDA #$ØØ
DØ STA $F1
D2 LDA $CØ8C,X
D5 BPL $BBD2
D7 ROL
D8 STA $FØ
DA LDA $CØ8C,X
DD BPL $BBDA
DF AND $FØ
E1 STA $ØØF8,Y
E4 EOR $F1
E6 DEY
E7 BPL $BBDØ
E9 TAY
EA NOP
EB CLC          good result
EC RTS
ED SEC          bad result
EE RTS
```

What happens is that the code from $BBØØ to $BB79 searches the disk for a certain sequence of bytes. When it finds them, it loads the next 8 bytes after the sequence into memory locations $FØ-$F7. $BB7A to $BB97 uses these bytes to decode the next stage of the boot code at $B7ØØ-$B9FF. After decoding, the next boot stage at $B7ØØ is JuMPed to at $BB95. What we have to do here is to load the correct values into $FØ-$F7 and then jump to the decode routine. The following code will do that nicely:

```
A9 FC     LDA #$FC
85 FØ     STA $FØ
A9 EE     LDA #$EE
85 F1     STA $F1
A9 EE     LDA #$EE
85 F2     STA $F2
A9 FC     LDA #$FC
85 F3     STA $F3
A9 E7     LDA #$E7
85 F4     STA $F4
A9 EE     LDA #$EE
85 F5     STA $F5
A9 FC     LDA #$FC
85 F6     STA $F6
A9 E7     LDA #$E7
85 F7     STA $F7
4C 7A BB  JMP $BB7A
```

I discovered what the correct bytes were by breaking into the monitor early in the boot process and displaying what was in $FØ-$F7. After making these changes to track $ØØ sector $Ø5, I found that the program (to my knowledge) works perfectly.

I hope that this softkey will help out those of you who are just starting out in the art of deprotection. It is a fascinating hobby and you really learn a lot from your computer, contrary to what many people say.

Step by Step

1 Boot up DOS 3.3 and disable error checking:

POKE 47426,24

2 Copy the BDCS disk:

RUN COPYA

3 On the copied disk, make the following changes:

Trk	Sct	Byte(s)	From	To
$ØØ	$Ø5	$ØØ-?	?	A9 FC 85 FØ A9 EE 85 F1 A9 EE 85 F2 A9 FC 85 F3 A9 E7 85 F4 A9 EE 85 F5 A9 FC 85 F6 A9 E7 85 F7 4C 7A BB

A. Evans

A.P.T. for...

Moebius

Origin Systems

■ Requirements

□ Sector Editor
□ *Moebius Realm* disk (sides C & D)

This APT is contributed in response to Dennis Gaunt's request in COMPUTIST #52 and also as a guide to those of you who are still struggling with this game.

It seems Origin Systems is consistent in its design of games where detailed character information resides on an unprotected second disk. With this in mind, I found saved character information on disk sides C and D (depending on which realm your character was last saved).

1 Boot *Moebius* and make note of your character name(s) and the realm(s) associated with each character "on an adventure".

2 Remove *Moebuis* disk and boot your sector editor.

3 Search *Moebuis* disk side C (Earth and Water realms) or side D (Air and Fire) for the sector that contains the name of your character. (I have found some of my character names at various locations such as tracks $16, $17, $1A, $22, $23, and $26 with varying sector locations.)

4 Use the following table to enhance your character. Maximum value at these locations is 255 ($FF) unless noted by '*'.

If, for example you found the character name that you were looking for on track $17, sector $Ø5, you would find the following information:

ATTRIBUTE	BYTE
Name	$ØØ-$ØF
Experience	$12,$13 ($13 * 256 + $12)
Level	$14
Body points	$15 (limit), $16 (current)
Mind points	$19 (limit), $1A (current)

Possessions and magic items are found on the next consecutive sector. (IE. Track $17, sector $Ø5 — track $17, sector $Ø6 or track $16, sector $ØF — track $17, sector $ØØ.)

POSSESSIONS	BYTE	MAGIC ITEMS	BYTE
Food	$44	Tiger teeth	$64
Water	$45	Beetle pincers	$65
Torches	$46	Soil sample *	$66
Whetstones	$47	Fish scales	$67
Body Elixirs	$48	Condor feathers	$68
Mind Elixirs	$49	Panda hair	$69
Shovel *	$4A		
Amulet *	$4B		
Lives	$4E		
Shurikens	$4F		

* Value should be $Ø1 if you have it, or $ØØ if you don't.

Paul R. Wilson

A.P.T. for...

H.E.R.O.

Activision

Using the crack for *H.E.R.O.* on page 22 of COMPUTIST #52, before doing step 6, add the following self modifying code patch.

```
3ØØ:A9 38 2Ø A8 FC AD ØØ CØ C9 DØ FØ F9
3ØC:C9 C6 DØ Ø8 A9 Ø5 8D 8D 6Ø 4C E2 6Ø
318:C9 D3 DØ Ø5 A9 ØØ 8D B5 6Ø 4C E2 6Ø
6ØB4:2Ø Ø5 Ø3                  was 2Ø E2 6Ø
```

Then do step #6 and add

BSAVE PATCH,A$3ØØ,L$28

Be sure to BLOAD PATCH along with *H.E.R.O.* when loading the game into memory.

Now *H.E.R.O.* has three additional keyboard commands.

F - normal speed
P - pause (missing in original game)
S - slow speed (essential for some very tight spots that almost always take a game Life when negotiated at normal speed)

Readers Data EXchange

Playing Tips for...

Castle Wolfenstein

Muse

1 Copy your cracked disk and label the "original" Disk 1. Label\the copy Disk 2.

2 Play on disk 1 to a good point (Like you have 1Ø bullets, 3 grenades, a vest, a uniform and the war plans and swap to disk 2. Press ESC to "save" (there is no true save feature on the game and if you resume and get caught or shot on the resumed game, it's back to room 1 with only 1Ø bullets. Worse, if you blow yourself up with a grenade, you lose the entire castle map! If you were playing a terrific game and getting near the exit, bye-bye progress and all your maps are now waste paper as the map is re-randomized.)

3 Resume on disk 1 and upon finishing the room, swap to disk 2 and exit. Play until you get caught or killed. Resume on disk 1.

4 Now comes the fun part! You will have all the ammo and stuff you started with, but your damage is still there! Rooms full of dead men, rifled trunks, grenade holes (rare) and only an occasional S.S. man will show up. Hold them up with a gun or grenade. In this reality warp, bullets shot or grenades used, or even the loss of vest and disguise will be repaired, but the damage (progress) remains. You can resave to disk 1 to avoid a long sequence of leaned out Nazi rooms (and Confronting S.S. men), and it'll only be a matter of an hour at most before you reach the coveted exit, and perhaps a promotion!

5 If a room proves intractable, siccing an S.S. man on you before you can even get into it, warp reality by swapping to disk 1 before entering this room.

WARNING: Switch back to disk 2 should that replacement room get too hot. Lest it get saved to disk 1 as you get offed and your progress is erased.

This A.P.T. requires a new line of strategy, but it makes *Castle Wolfenstein* playable, instead of frustrating.

Sometimes you may suffer a drop in rank should you get shot. This is not fair and you should use A.P.T.s from COMPUTIST to undo this.

Remember:
Disk 1 - Boot and Start Game on this
Disk 2 - Play game on this

At higher levels, vest wearing Nazis increase in frequency.

I've gotten to level 6 this way.

Steven Heckler

I have a technical question concerning the Apple IIc. How do you run quarter tracked programs like *Karateka* (original version), *Choplifter*, *Starblazer*, and the old *One-on-One*?

I noticed in your listing of back issues that you had an article titled "Playing Karateka on a IIc (COMPUTIST #37, I believe). I know I must sound like a mooch since I do not subscribe to your magazine, but no one I have talked to has had any idea how to run the quarter tracked programs on the IIc.

According to Christopher Dean (COMPUTIST #52) the problem with II plus and IIe programs that don't boot on a IIc is caused by the different BootØ code in ROM. Programs that use the ROM BootØ code directly will not work on the IIc without modifications. He goes on to show how to make these changes using Drol, Hardball, Orge and Arctic Fox as examples. ....RDEXed

James E. Bulman

As a new subscriber to COMPUTIST, I would like to express my thanks for an excellent magazine. I was especially pleased with my first issue, COMPUTIST #52. With it, I was able to copy two of my copy protected programs (*Ultima I* and *Shanghai*). With that said, I would like to offer some constructive criticism.

1) Clearly list the softkeys in your back issues. I ordered several back issues only to find the softkeys were for the Apple IIgs (I have a IIc). Softkeys in back issues that require information from other back issues should also be noted. (The *Carmen Sandiego* softkey in COMPUTIST #25 requires information from COMPUTIST #19. I have ordered #19 praying that the modified F8 ROM is NOT a hardware modification!)

2) When softkeys in current issues refer to previous issues, re-list the information from those issues. (The softkey for *Math Blaster* in COMPUTIST #53 is an example.) I realize this would cut down on back issue orders but it would also reduce reader frustration.

Thank you again for a fine computer magazine. I am learning more and more with each issue.

The modified F8 ROM is hardware, but you could get the same effect using Senior PROM on your IIc.

We usually don't reprint softkeys because of the wealth of new material that arrives here every day. It is difficult to justify reprinting old material when the next issue is already full and letters recieved today must wait for the issue after next. Our focus is to get your letters printed as quickly as possible. ....RDEXed

Alex Lee

Playing Tips for...

Conan

- The bat on the first board can be killed without losing a sword.
- Stay on the bubble for as long as you can.
- Collect swords and a jewel on the fourth level.
- You have to kill six dragons before the door will unlock on the fifth level.
- The eyes will help you build a ladder on the sixth level.
- If you run into trouble on the seventh level, you can always drop back one.

I have a suggestion. How about someone starting a beginner's column. You could explain some of the more common protection schemes, how to find what protection is being used and how to break the schemes once you find them.

I have a Pineapple brand Apple compatible. Several keys can be pressed down but nothing is registered in the CPU the broken keys are: 1, ESC, Ctrl-A, 5 (on the numeric keypad) and shift Z. Does anyone know if this can be fixed or, if it's not fixable, where I can get a replacement keyboard? My address is P.O. Box 265, Claverack, NY 12513

David M. Widman

Softkey for...

Hacker II: The Doomsday Papers

Activision

■ **Requirements**

□ Apple II series with 64k (minimum)
□ *Hacker II* disk
□ A blank disk
□ COPYA

Hacker II is a very good program, as well as *Hacker*. This time the government needs a little favor; as a computer wizard you must hack your way through Siberian hazards in a maximum security complex. About the protection, it is almost the same as *Hacker*, there is a subroutine that checks track $ØØ from the disk and puts a FF at location $FC and a 55 at another special location that is verified by the program.

Step-by-Step

1 Copy the *Hacker II* disk with *COPYA*.

2 Put your *COPYA*ed disk in drive 1 and enter:

```
BLOAD HACKER II HELLO,D1
CALL-151
63C9:A9 FF 85 FC A9 55 4C 1Ø 64
BSAVE HACKER II HELLO,A$6ØØØ,L$589
```

You are done!

Softkey for...

Labyrinth

Activision

■ Requirements

- □ Apple IIe or IIc with 128k (minimum)
- □ *Labyrinth* disk
- □ A blank disk
- □ COPYA

Labyrinth is an animated adventure game with double hi-resolution animation and a very attractive scenario. About the protection, it is almost the same as *Hacker* and *Hacker II*, there is a subroutine that checks track $ØØ and puts an FF at location $FC and puts a 55 at another special location that is verified by the program.

Step-by-Step

1 Copy the two sides of the *Labyrinth* disk with *COPYA*.

2 Put your *COPYA*ed disk side one in drive 1 and enter:

```
BLOAD BOOT2,D1
CALL-151
8B9:A9 FF 85 FC A9 55 4C ØØ Ø9
BSAVE BOOT2,A$8ØØ,L$279
```

Now you have a *COPYA*-able version of *Labyrinth*.

UNK, Prince George, Canada

Softkey for...

Arctic Fox

Electronic Arts

After reading my first issue of COMPUTIST which was #53, I found a much easier way of deprotecting *Arctic Fox* than doing all the sector edits proposed in Scott M. Simon's softkey. After reading the softkey for *Earth Orbit Station*, I noticed that it also, when copied, showed many errors in track $Ø6. Since *Arctic Fox* was also an EA disk, I tried it. To my relief, it worked. Since I had already typed in *Earth Orbit Station* it was much easier than doing all those sector edits. I wouldn't be surprised if it worked on other EA wares. Many thanks to Bob Wilson! For any of you who missed Bob Wilson's controller, here it is.

Controller

```
1ØØØ REM EOS/ARCTIC FOX
1Ø1Ø TK = Ø:LT = 35:ST = 15:LS = 15:CD = WR:FAST
     = 1:MB = 55
1Ø2Ø GOSUB 49Ø: GOSUB 61Ø: IF TK = 1 THEN T1 =
     TK:TK = PEEK (TRK): GOSUB 31Ø:TK = T1
1Ø3Ø GOSUB 49Ø: GOSUB 61Ø: IF PEEK (TRK) = LT
     THEN 1Ø5Ø
1Ø4Ø TK = PEEK (TRK):ST = PEEK (SCT): IF TK = 6
     THEN TK = 7:MB = 151
1Ø45 GOTO 1Ø2Ø
1Ø5Ø HOME : PRINT "COPYDONE" : END
5ØØØ DATA 6^CHANGES
5Ø1Ø DATA 1,1Ø,82,24
5Ø2Ø DATA 1,1Ø,83,96
5Ø3Ø DATA 1,1Ø,84,72
5Ø4Ø DATA 1,15,Ø,24
5Ø5Ø DATA 1,15,1,96
5Ø6Ø DATA 1,15,2,221
```

Controller Checksums

```
1ØØØ  - $356B      5ØØØ  - $3661
1Ø1Ø  - $99DD      5Ø1Ø  - $5711
1Ø2Ø  - $17C5      5Ø2Ø  - $326Ø
1Ø3Ø  - $5B83      5Ø3Ø  - $5F61
1Ø4Ø  - $24C3      5Ø4Ø  - $ØAEA
1Ø45  - $A2ØØ      5Ø5Ø  - $5B72
1Ø5Ø  - $9FA4      5Ø6Ø  - $C15F
```

Jim S. Hart

Softkey for...

Math Blaster

Davidson and Associates

■ Requirements

- □ *Math Blaster* original disk
- □ 2 blank disk sides
- □ *SUPER IOB* v1.5 with both NEWSWAP. CON and FAST.CON controllers (from Starter Kit)
- □ *Word Attack/Classmate* softkey from COMPUTIST #28 for reference

■ Optional

- □ "Beneath Apple DOS" for an explanation of what format changes are
- □ A fast DOS such as *Diversi-DOS* or *ProntoDOS* (not necessary but helpful)

Math Blaster is a user-friendly math tutorial and drill that emphasizes the fundamentals. Students and teachers alike have applauded the effort that Davidson & Associates put into making the programs as well written as possible. They came up short, however, in the ease of making backups. Parents, once finding out that the disks cannot be copied, usually will not let their kids use the program unless they are around for fear of the disk crashing. Never fear, COMPUTIST is here to alleviate your worries (and keep money in your pocketbook). Below are the steps to follow to get rid of the annoying format protection along with a nasty secondary protection scheme.

1 Boot up DOS 3.3, initialize the two blank disk sides, and delete both hello programs. I have only tried *Diversi-DOS*, *ProntoDOS*, and standard DOS 3.3, so you take your chances if you use some other type.

```
INIT HELLO
DELETE HELLO
```

2 First of all, we must convert the *Math Blaster* disk to normal format. The easiest way to do this (usually) is to capture the perverted disk's RWTS, and then use the NEW SWAP controller with *Super IOB v1.5* to read the disk in with it's own RWTS and then write it out to a normal disk using normal RWTS.

The problem is how to get the RWTS. It turns out not to be too hard. Boot up the *Math Blaster* disk and when the Applesoft prompt (]) appears, open your disk drive door. There will be some rattling and then you get an I/O ERROR message. At this point, I discovered that you cannot type in any commands. This means that the BASIC RUN flag at $D6 has been set to a value greater than 127. This causes BASIC to interpret all commands typed in at the keyboard as 'RUN'.

A little known fact is that DOS commands still work. The DOS command 'FP' resets all pointers to their defaults, even the BASIC RUN flag!

Ah ha! If you try this, you will discover that the DOS commands have been altered a la Beagle Bros *DOS BOSS*.

To make a long story short, I went a-huntin' through memory via a copy card and discovered that FP had been renamed PF. Type this in and you can now execute all other commands such as CATALOG, LIST, etc:

```
PF
```

3 Now we need to move the RWTS to a safe area so a normal DOS disk can be booted. Move the RWTS and then boot the blank initialized disk:

```
CALL -151
19ØØ<B8ØØ.BFFFM
```

Insert the blank initialized disk.

```
C6ØØG
```

4 Save the RWTS.

```
BSAVE RWTS.MATH,A$19ØØ,L$8ØØ
```

5 Load *SUPER IOB V1.5* and install the NEW SWAP controller. Then add the following lines to copy *Math Blaster*.

```
LOAD SUPER IOB
EXEC NEWSWAP.CON
1015 TK = 3
1ØØ1Ø PRINT : PRINT CHR$ (4) "BLOAD
    RWTS.MATH"
RUN
```

6 Now clear memory, load *SUPER IOB V1.5* again, install the FAST controller into it, add two lines, and RUN the result to copy the *Math Blaster* data disk onto the second blank initialized disk:

```
FP
LOAD SUPER IOB
EXEC FAST.CON
1Ø2Ø POKE 47426,24 : GOSUB 49Ø : GOSUB
    61Ø
1Ø3Ø POKE 47426,56 : GOSUB 49Ø : GOSUB
    61Ø : IF PEEK (TRK) = LT THEN 1Ø5Ø
RUN
```

(Note: this altered FAST controller can convert a surprising number of older educational disks into normal format.)

7 Put away your original *Math Blaster* disk. We're done with it.

8 Boot up normal DOS and then insert the copied *Math Blaster* program disk into the drive.

9 Rename their boot program so we can add our own:

```
RENAME HELLO, MATH
```

10 Type in the one line BASIC program below and then save it to the *Math Blaster* program disk:

```
1Ø TEXT : HOME : HTAB 15 : PRINT ""MATH^BLASTER''
    : PRINT : POKE 47721,96: POKE 47741,96:
    POKE 4Ø222,87: POKE 4Ø223,255: PRINT CHR$
    (4) ""RUN^MATH''
```

```
SAVE HELLO
```

11 Write protect the disk and you are done!

For the adventurous (or curious) among you, I suggest reading Dave Stanton's "Word Challenge/Classmate" softkey in COMPUTIST #28. He talks about programs hidden within programs which is exactly what the situation with *Math Blaster* is.

Throughout the programs, there is a nasty secondary protection that will, upon discovering normal DOS 3.3 present, initialize your disk. Calls are made to the start of the DOS 3.3 INIT routine which in the *Math Blaster* DOS has a Return from Subroutine (RTS) command. I found this out quickly when, upon first booting the supposedly softkeyed disk, it reinitialized the disk! Other checks are made and if things are not what they are 'suppposed' to be, an indirect call is made to the INIT command.

I found a few occurrences of these two checks, but since there were programs hidden within programs, I decided to fix the INIT routine in normal DOS so that it would mimic *Math Blaster*'s original DOS.

This entailed putting two RTS's at the INIT entry points (line #4Ø in our new HELLO program) and changing the pointer which the second check uses to call INIT into a harmless call to monitor RTS at $FF58 (line #5Ø). For a simple lesson, copy the softkeyed *Math Blaster* program disk onto another blank disk which we'll call the "DUMMY' disk. Now DELETE the HELLO program on the DUMMY AND RENAME MATH, HELLO. This returns the disk to it's original volatile state. Boot this disk and listen closely. BINGO. The disk reinitializes itself!

The moral to this is to write protect any newly softkeyed disk to prevent much frustration until you are absolutely sure there is no secondary protection lurking around.

Enjoy your deprotected *Math Blaster*!

Softkey for...

Tuesday Morning Quarterback

Automated Simulations

Requirements

- □ 64K Apple II+ or the ability to RESET into the monitor at will
- □ *Tuesday Morning Quarterback* original
- □ A blank disk
- □ File copy program such as FID
- □ Fast DOS (not necessary but helpful)

When the football season rolls around, I get to watch plenty of action on the field. Football is my favorite sport and I can never watch enough of it. After the season is over and football is no more for 4-5 months, I satisfy my football cravings with Automated Simulations' *Tuesday Morning Quarterback*.

It is an excellent simulation even though the screen graphics are a bit slow and sparse. The game cannot really be enjoyed, however, unless there is no chance of the original crashing. This is so you will not have to wait for the backup to arrive via the mail (which takes time and $$$).

Deprotecting the disk is one way to prevent those fears. Follow along with the steps in this article and when you are done your program will be *COPYA*-able which translates to "No more hassles!".

NOTE: Folks who have the ability to reset into the monitor should ignore instructions pertaining to the 16K RAM card.

1 Initialize the blank disk, preferably with a fast DOS such as *ProntoDOS* or *DiversiDOS*, and delete the hello program:

```
INIT HELLO
DELETE HELLO
```

2 After the blank has been initialized, drop down into the monitor and set up the 16K RAM card so that it contains an image of BASIC and the monitor in it. We also need to change the reset vector (i.e. where the computer goes when the reset key is pressed) so that it points to the monitor. This emulates those Non-Maskable Interrupt cards that allow you to go into the monitor at will. Capturing the protected (and encrypted) Applesoft files requires that we have this ability. For more information regarding RAM cards, check out Wes Felty's "More ROM Running" article in COMPUTIST #34.

```
CALL-151                  go into monitor
CØ81 CØ81         assumes slot Ø RAM card
DØØØ<DØØØ.FFFFM           move ROM image
FFFC:59 FF           change reset vector
CØ8Ø                     enable RAM card
```

3 Insert the *Tuesday Morning Quarterback* (*TMQ* from now on) original into drive #1 and boot it:

```
C6ØØG
```

4 When the first screen comes up, press RESET to go into the monitor.

5 What we must do now is move part of the Applesoft program to a safe place so that a boot will not wipe it out. This involves checking out where the program end is, writing it down for future reference, and then doing the actual memory move. First let's see where the program end is:

```
AF.BØ
```

Write down the two hex digits that show up after you press RETURN - we'll need them in a moment. Now move the section of the Applesoft code that would be destroyed by a boot to a safe place:

```
7ØØØ<8ØØ.8FFM
```

6 Now boot up the blank initialized disk. You will get a "FILE NOT FOUND" error when there isn't a program named 'HELLO' found on the disk. For right now that is fine because nothing will be loaded into memory which would cause us to move more memory (yuch). We now have to move the beginning of the program back in place and then inform DOS and BASIC of its length:

```
8ØØ<7ØØØ.7ØFFM
AF:xx yy    put the two bytes you wrote
          down earlier in place of xx and yy
3DØG
```

7 We will now save the program to disk then set the computer back up so that another protected BASIC file may be loaded.

```
SAVE STARTUP
CALL -151
CØ8Ø
```

insert the TMQ original disk into the drive

```
C6ØØG
```

8 For the rest of the BASIC files on the *TMQ* original you will have to follow the steps from 5 to 7 using the different file names and RESETing into the monitor at the different times shown in the list below.

The list order is — Filename — end of program value (from $AF-BØ) — when to press **RESET** —.

For example, to get the file "MENU", you have to wait until a menu showing the different choices, such as play football or practice kicking, comes up before you press **RESET** to go into the monitor. From there on follow steps 5 to 7 and save the file with the name "MENU" in place of "STARTUP'.

9 When you are done with the above steps, you will have five BASIC programs on the copy disk with the names STARTUP, MENU, KICK PRACTICE, TEAM FILE EDIT, AND TUESDAY MORNING QUARTERBACK. Now the text files, containing the team's information, must be copied from the *TMQ* original to the copy. We are fortunate that Automated Simulations decided not to protect these files. FID, or any file copier, should be used now to copy the text files named NFL.DATA, SUBS, and the TEAM## (## = the numbers from 1 to 3Ø, inclusive) files to the copy. When you finish copying these text files, there should be five BASIC programs and approximately 32 text files on the copy disk.

10 The final thing to do is to rename the STARTUP file so that it will automatically run upon booting:

```
RENAME STARTUP, HELLO
```

You now possess a deprotected copy of *Tuesday Morning Quarterback*. The files are now open for inspection and customization. Enjoy!

List #1

STARTUP (EF Ø8) at first screen.

MENU (B8 Ø9) when menu with 'PLAY FOOTBALL' and 'PRACTICE KICKING' options appear. Shows up right after the first screen.

KICK PRACTICE (49 18) Go to menu and choose 'PRACTICE KICKING' option. Go into monitor when drive stops.

TEAM FILE EDIT (CD ØE) Go to menu and choose 'UPDATA TEAM ROSTERS' option. Go into monitor when drive stops.

TUESDAY MORNING QUARTERBACK (FØ 6C) Go to menu and choose 'PLAY FOOTBALL' option. Go into monitor when drive stops.

Dr. Leigh Rowan-Kelly

I sent a short version of the method to remove the protection from *Softswitch* and a few other programs. I have now found that, with regard to *Softswitch*, my version was grossly inadequate. Another disk had the code in another place, so I have now expanded my coverage of the softkey to include a full explanation.

Softkey for...

Softswitch

Roger Wagner

Essentially *Softswitch* is an outstanding program, made poor by virtue of some very sneaky protection. It writes a special byte to the battery backed-up RAM (at address $FB) the first time you install the program on any disk using your system. This RAM has been designated as one of the reserved areas by Apple. When you boot a disk containing *Softswitch*, the program then checks for this special signature byte in the RAM, and won't install unless the low bits of the byte are "1Ø", or else it requires the original protected disk to be on line somewhere.

What this means in simple terms is that if you replace or disconnect your battery, or have your Motherboard replaced during repair or upgrading, you lose that signature byte and *Softswitch* hence will then work only with the original disk online. Given the additional complication that *Softswitch* will only install once (by inserting that stupid byte), then effectively you have a useless disk and program!

It is possible, however, to make an alteration to the file /SS.SYSTEM.DISK/SYSTEM/SYSTEM.SETUP/TOOL.SETUP.2 to correct this. You can BLOAD this file at $2ØØØ and by typing `48 A2 Ø3 ØC 22 ØØ ØØ E1 68<2ØØØ.88ØØP` you can locate the code that reads the byte. It was at $2BA9 on this disk. If you then replace that code with `48 A2 Ø3 ØC 68 68 A9 FE ØØ` and then BSAVE the file back to the disk (naturally using a copy of your original!), then *Softswitch* no longer checks for the signature byte in the battery-backed RAM. You can even copy the file to other disks and have them install *Softswitch* in memory without going through the full installation procedure.

If you can't be bothered doing it this way, then you can, of course, simply search the disk (using *Copy II Plus* v8.2 or *Zap* from *Bag of Tricks 2* or any sector editor with search facilities) for that same sequence of bytes and change it accordingly. I found it at block $112, starting at byte $3D on one disk, and at block $11D, byte $4A on another. I might add that this was based rather heavily on a letter that appeared in Open-Apple, and I am indebted to Peter Stubbs for writing that letter.

The letter in question is in the September issue of OPEN-APPLE, a publication that we heartily recommend. One year subscriptions are available for $28 from; Open-Apple, P.O. Box 1125Ø, Overland Park, KS 662Ø7

. . . . RDEXed

Softkey for...

Hardball gs

Accolade

I see you have another crack for *Hardball gs* in COMPUTIST #53. My edits also seem to produce a working copy, but I only used the two byte changes, at block $32C, bytes $1EE-1EF (from C2 2Ø to 18 6Ø) and block $32F, bytes $FF-1ØØ (from DA C2 to 18 6Ø).

Anyway, keep the softkeys coming.

Jan Recourt

Playing Tips for...

Ultima IV

Origin

I have the solution regarding the Stigian problem I asked about. When you are in the big room with the altar in the middle, use stones. After that answer the questions and the rest is simple.

(?) Is there anyone who can give me more lives in *Moebius*?

(?) Why can't I put Baudville's *Video Vegas* on a hard disk or 3½" disk with *Unidos Plus* by Microsparc. This seems to be the same problem with *Ultima IV* and *Auto Duel*.

When you put them on a 3½" diskette after the startup, it searches in the 5¼" drive for the rest of the game. I think there must be a change for the drive searching. It would be nice to have *Ultima* or *Moebius* on one 3½" diskette.

Alan Sheppard

I have recently inquired about backing up *EDD 4* v4.4 and was advised that COMPUTIST #49 had the program required. I was also told that the library disk contained the program as well. Because of this I ordered both the disk and magazine and started a one year subscription.

I have some problems and need a little help.

Readers Data EXchange

I booted up library disk #49 disk and could not find any program for *EDD-4*, as well, 11 of the 14 or 15 programs on the disk locked, gave me syntax error or indicated undefined statement. Is there something I am doing wrong?

Going back to the manual, I tried typing in the program for version 4.4 and initially thought I was successful. On the slave disk the Hello program came up with EDD-4 as well. I would then select EDD-4 and it would work fine, that is until I cold booted the disk. At first it appeared to work fine, a color logo came up, I pressed the space bar and the next screen (credits) appeared. The next press of the space bar would put me into the main program, however, when I did this the screen partly filled with characters and my cursor showed me that I was in BASIC. Can you help me with this one?

The Starter Disk seems very good but I am wondering if it is possible to print out the documentation.

As you can see, I am a novice, anything you can do to help me will be appreciated.

Could I also get a copy of what softkeys are on all your library disks?

Thanks.

The material we put on our library disks consists of lengthy Applesoft programs, controllers, hex dumps, etc. In other words, material that would be a real pain to type in. Step-by-step procedures must still be typed in. Sorry about the misunderstanding.

The reason you are getting error messages is, you are probably trying to run a controller on its own. Controllers can be easily identified because their names begin with "CON.". A controller, as the name implies, is the control portion of SUPER IOB, a program on the Starter Kit. The controller tells SUPER IOB what to do and when to do it. Since it calls subroutines within SUPER IOB, a controller cannot be RUN on its own. It must first be merged with SUPER IOB and then RUN.

Many of the binary programs (type B files) on our library disks are also part of other programs and are loaded or run by those programs. You shouldn't try to BRUN them directly. Read the COMPUTIST issue carefully to see which programs may be RUN/BRUN and what other programs may be required.

The softkey for EDD 4.x is of a type called "boot code tracing". This is a "storm the walls" approach, where the program in question is followed as it loads into the computer. Detailed notes are kept and, step-by-step, each portion of the program is allowed to load. At the appropriate moment, when the program is in memory but not yet activated, the entire code is saved to disk as one or more files. This type of softkey is usually dependent upon a particular version of a program. If the softkey doesn't work for you, there are three obvious paths to take. Carefully redo the softkey and hope you did some step incorrectly the first time. Write the author and hope that individual has time to spare to work on your problem. Or, write to RDEX with your question and let thousands of readers in on the problem. Be sure to give all required information, including the version number of the program and all steps that you have already tried with what results. Someone may already have solved that particular problem or be working on it at that very moment.

As to printing the Start Kit documentation, the easiest way is if you have a text editor that will load the "DOC." files. Then you could delete the control characters and reformat for 80 columns. If you have Copy II Plus, you could use the view files option, select text and specify the "DOC" files then answer "Y'es to the printout question.

We don't have a list of what is on each library disk but, if you are primarily interested in controllers, the best deal is to get the Super IOB collection. It contains all of the controllers thru COMPUTIST #38. (125 in all.) .... RDEXed

Michael A. Mahaffey

I recently subscribed to your magazine and I have found it excellent, but only half of what I am looking for. I am very interested in finding how to deprotect my software because I hate to work with the original disk for fear of something happening to it. Your magazine is just the thing for such a purpose and very good as far as that goes. However, I am looking for something else. The other half of what I want is to be able to put deprotected programs on another disk with other programs. Thus reducing my disk count, increasing convenience, saving time and reducing desk clutter. Since I have an Apple IIgs, I want to put all my programs on 3½" disks.

Have you, in previous issues, shown how to transfer programs from 5¼" (DOS 3.3) to 3½" (ProDOS) disks? If not, do you know where I can find such information? If this sounds like a naive question, please bear with me. I am a previous TRS-80 aficionado recently converted to Apple and I am struggling with the differences.

Also, I have seen a couple references to *Senior Prom*. Can you explain what it is and where I may obtain one.

In order to transfer programs from 5¼" disks to 3½" disks, the programs must be in normal format files on a non-protected disk. Many of the softkeys in COMPUTIST show you how to do this. Even then, incompatibilities exist that limit the transfers. Some files require a specific DOS, others make direct read/writes from specific block/sectors. Sometimes it is something as simple as hard coding the drive number or prefix name. So, to answer your question, we show you how to transfer files. If you have a specific program in mind, please name the program and address your question to RDEX for maximum response.

The Senior PROM is a hardware (firmware) device that allows a IIc/e to interrupt a running program. It contains many useful utility programs that enhance your ability to manipulate programs in memory and to do disk modifications. It is available for $79.95 from Cutting Edge Enterprises, 43234c Ren Cen Station, Detroit, MI 48243.......RDEXed

David Sheppard

Softkey for...

Wortgefecht

Gessler Educational Software

■ **Requirements**

□ Apple II 48K
□ FID

This is a German vocabulary disk that has been protected by the use of alternating address headers on every other track.

1 Initialize a disk with your favorite fast DOS using HELLO as the startup program.

2 BRUN FID from your Apple system master.

3 Enter (9) to quit.

4 Enter the monitor.

```
Call-151
```

5 Enter the following:

```
B942:18                        dissable errors
BE48:18                       dissables errors
B954:29 00    ignore 1st address header byte
803G                               restart FID
```

6 Place *Wortgefecht* in drive 1 and copy all the files to your initialized disk. That's all! It's now *COPYA*-able.

Thanks for the great magazine.

Thomas Purifoy

I just received the copy of my article in COMPUTIST #54 (*Might and Magic*, page 21) and I noticed a few mistakes in it. I am afraid that some of my calculations were printed wrong. First, the numbers in the unmarked column should be 16 with an exponent. The exponent explains how the 16 is to be used in multiplying the different nibbles.

```
$27 1st nibble  16's place (16^1)
    2nd nibble  1's place (16^0)
$28 1st nibble  4,Ø96's place (16^3)
    2nd nibble  256's place (16^2)
$29 1st nibble  1,Ø48,576's place (16^5)
    2nd nibble  65,536's place (16^4)
$2A 1st nibble  268,44Ø,ØØØ's place (16^7)
    2nd nibble  16,777,216's place (16^6)
```

My example of the math part of the APT was nearly unreadable. The numbers should have been printed in a column like this:

i.e. You have 1,694,834 experience points, thus:

```
byte $27 = $72            7 x 16 = 112
             ↑↑           2 x 1  = 2
     (7x16)(2x1)
byte $28 = $DC        13 x 4Ø96 = 53248
             ↑↑       12 x 256  = 3Ø72
   (13x4Ø96)(12x256)
byte $29 = $19     1 x 1Ø48576 = 1Ø48576
             ↑↑    9 x 65536   = 589,824
 (1x1Ø48576) (9x65536)
                        total = 1,694,834
```

I really like the new RDEX, keep up the good work.

Sorry for the confusion. The way a file is typeset is strongly dependent on which volunteer is sitting at the typesetter when your file comes across. Some people are just a little more experienced with the machinery and it shows. .... RDEXed

Scott Lloyd

Softkey Addendum for...

Ikari Warriors

Data East

When I was modifying the *Ikari Warriors* for unlimited men, bullets and grenades, the sector edits were not where they were listed in COMPUTIST #53. I found them in sector $ØE instead of sector $Ø9.

On a side note, the place where I work has a big problem with changing to side B on the disk. With the protection removed, the disk works fine although I had to edit my disk for "unlimited everything" to find this out.

Carl D. Purdy

I have unlocked several pieces of software which I would like to share with you at this time. These are Milliken math series, copyright 198Ø, the Micro-Computer Learning games from Learning Well Co., Morning Star Math, and Reading Comprehension Main Idea and Details from Milton Bradley.

Softkey for...

Milliken Math series

Milliken

■ **Requirements**

- □ *Demuffin Plus*
- □ A way to break into the monitor
- □ DOS 3.3 or a Fast DOS (*Pronto DOS* from Beagle Bros.)

1 INIT a blank disk with DOS 3.3 or a fast DOS. Have a slave disk handy to save the RWTS.

```
INIT HELLO
```

2 Insert the Milliken disk and boot it.

```
PR#6
```

3 Break into the monitor and move the RWTS to a safe location.

```
19ØØ<B8ØØ.BFFFM
```

4 Insert a slave disk and boot it.

```
C6ØØG
```

5 Insert your *Demuffin* disk and save the Milliken RWTS.

```
BSAVE RWTS.MILLIKEN,A$19ØØ,L$8ØØ
```

6 Fiddle with DOS and use *DEMUFFIN PLUS* to copy some files.

```
BLOAD RWTS.MILLIKEN,A$68ØØ
CALL - 151
B8ØØ<68ØØ.6FFFM
8Ø3G                          Starts Demuffin
```

From here follow the prompts to save the files on the Milliken disk to your INITed disk. The files are as follows:

```
BOOT
A.LOGO
MAT.D
MANAGER
CHR SETS YZ
INIT
SD.ADDITION  (SD.MULTIPLICATION,
   SD.DIVISION, SD.SUBTRACTION)
B.TEXT POINTERS
B.TEXT
ANIMATION
DIVISION  (MULTIPLICATION, DIVISION,
   SUBTRACTION)
B.ALL 9
```

7 Once all the files are copied, rename the BOOT program "HELLO" and you are finished.

```
RENAME BOOT,HELLO
```

Softkey for...

Micro-Computer Learning Games:
Race Track/Reading for Detail
Time Capsule/Reading Skills
Chief of Detectives/Drawing Conclusions
Mountain Climbing/Cause & Effect
School Days/Inference
Galaxy Search/Predicting Outcomes

Learning Well Co.

■ **Requirements**

- □ DOS 3.3 or a Fast DOS (*Pronto DOS* from Beagle Bros.)
- □ CopyA
- □ A file copy utility such as *Copy II Plus* or FID

The *Micro-Computer Learning Games* are a series of reading games that are really neat to use in the classroom, however, I hesitate to put originals in the hands of my students. Thankfully, it is quite simple to copy these.

1 Boot your DOS 3.3 system disk.

2 Tell DOS to ignore checksum and epilog errors and use *COPYA* to copy the protected disk.

```
POKE 47426,24
RUN COPYA
```

3 INIT a disk.

```
INIT HELLO
```

4 Use a copy utility such as *Copy II Plus* or FID to copy all of the files from the *COPYA* disk to your INITed disk.

5 Determine which file is the boot program. It will be something like BOOT, or BOOT1. Rename this file to HELLO.

```
RENAME BOOT,HELLO
```

Softkey for...

Reading Comprehension Main Idea & Details

Milton Bradley

■ **Requirements**

- □ A way to break into the monitor
- □ DOS 3.3 or a Fast DOS (*Pronto DOS* from Beagle Bros.)
- □ *Super IØB* and a *Swap Controller*

1 INIT a Blank Disk.

```
INIT HELLO
```

2 Boot *Reading Comprehension*.

3 Break into the monitor and move the RWTS.

```
19ØØ<B8ØØ.BFFFM
```

4 Boot a slave disk.

```
C6ØØG
```

5 Save the RWTS.

```
BSAVE RWTS.READING,A$19ØØ,L$8ØØ
```

6 Insert your *Super IOB disk*.

```
LOAD SUPER IOB
EXEC SWAP.CON
```

7 List line 1ØØ1Ø and change it to use your RWTS, i.e. RWTS.READING. Place slave disk in drive and run *Super IØB*.

```
RUN
```

8 Follow the prompts to copy the disk. DO NOT reINIT the disk.

9 When the copy process is done, boot the copied disk and catalog it. Determine which program is the boot program and rename it to hello. That is all there is to it.

Softkey for...

Morning Star Spelling

1 Boot your DOS 3.3 system disk.

2 Tell DOS to ignore checksum and epilog errors and use *COPYA* to copy the protected disk.

```
POKE 47426,24
RUN COPYA
```

No other changes are necessary.

Stephen J. Scalia

As my subscription ended last month, I thought I might drop you a few lines to tell you why I did not renew.

Although I am not a novice Apple user, I have owned my IIc for over four years now, I have yet to be able to utilize one softkey or feature program from your publication.

I have typed in almost 2ØØ programs from such magazines as "Nibble", 'Compute", 'Incider" and "A+". I have gotten all of the programs to run and have been able to modify them to suit my personal needs.

"Date/Time Without a Clock Card" by Steve Marvin in COMPUTIST #49 is an example of the problems I encountered. There is absolutely no way a non-assembly language programmer could decipher how to make this program run. Surely the appropriate changes for each version could have been published so us BASIC programmers could make use of the program.

I enjoy my Apple and try to subscribe to any publication that will broaden my knowledge. I must say that after 12 issues of COMPUTIST, I don't know anymore about copy protection than I did before I spent the $32.ØØ for the subscription.

Perhaps you are only interested in writing for people with a vast knowledge of assembly language but you are missing a lot of readers by not making your publication understandable to the 95% of us Apple owners who never will own an assembler.

I'll use my COMPUTIST subscription renewal money for something that talks my language. "Open-Apple" maybe.

Our writers are people just like you. Non-professionals who just want to let others know what they've done or something they've found out. Their letters are sometimes short and cryptic. But we're all friends here. If you need help with something, all you have to do is ask.
.... RDEXed

Dr. George Sabeh

I have enjoyed your publication over the past couple years. I have purchased most of the available back issues. It has helped me understand and enjoy my Apple. I would like to contribute two short softkeys which may help some of your readers.

Softkey for...

Ski Crazed
Baudville

■ **Requirements**

- □ *Locksmith Fast Copy* or a similar copy program such as *CopyA*
- □ Sector editor
- □ Blank disk

1 Copy the original using any copy program, such as *Locksmith Fast Copy*.

2 Sector edit the following:

Trk	Sct	Byte(s)	From	To
$ØØ	$Ø4	$A5-A6	DØ F7	EA EA
	$Ø6	$11-12	DØ F7	EA EA
		$2E-2F	DØ D5	EA EA
	$ØF	$Ø8-Ø9	DØ F7	EA EA

Make sure you write the sectors back to disk.

Softkey for...

Earth Orbit Stations
Electronic Arts

■ **Requirements**

- □ *Locksmith Fast Copy* or other copier that ignores errors.
- □ Sector editor such as *Copy II Plus*
- □ Blank disk.

1 Copy the original using *Locksmith* or similar copy program.

2 Sector edit the following:

Trk	Sct	Byte(s)	From	To
$Ø1	$ØA	$52-54	6C 54 ØØ	18 6Ø 48
	$ØF	$ØØ-Ø2	4C 69 AØ	18 6Ø DD
		$6F-71	4C 69 AØ	18 6Ø DD

Make sure to write the sectors back to disk.

These two softkeys would not have been possible if it was not for the help gained from reading the back issues of COMPUTIST. Keep up the good work.

Name withheld by request

Advertising Pirate BBS's and Encouraging Piracy

I am a COMPUTIST subscriber and I love the new RDEX, and ZOXCOPIES, however, COMPUTIST #53 has raised grave concerns. On page 37 you printed a letter from Chad Baker querying about the new user password to a large and infamous Pirate BBS in southern California called Motherboard West. Chad also advertises another BBS called Alien Nation. These boards exist primarily to trade cracked copies of software.

I hate copy protection and I have become an intermediate cracksmith through practice and some help from COMPUTIST. I am NOT A PIRATE and I fear that if you continue the practice of printing this type of letter your magazine will suffer from just, legal action from the Software Publishers Association. Chad, I assume, is a young man and you cannot blame him for his ignorance in making this type of query. However, I hope you will develop a standing editorial policy to prevent future incidents.

By the way, it might be a good idea to give Bill Bennett's BBS at 415-349-8245 a few words next month. Bill's board is dedicated to project STOP, the Software Theft Opposition Project. The board functions as a forum where interested parties can discuss the software piracy issue. Many pirates as well as rep's from Electronic Arts and other software publishers are currently active on the board. You may also write to Project STOP, P.O. Box 3142, San Mateo, CA 944Ø3.

The problem is that we just don't know which Bulletin Board Systems carry what kind of information. When it comes to BBSs, we aren't very active or knowledgable. We just print what you write. This whole magazine is just a readers data exchange. It would be frivolous for the SPA to pursue us on that issue. (No pun intended.) . . . . .RDEXed

Leh-Wen Yau

IIgs Softkey for...

Marble Madness
Electronic Arts

I purchased Electronic Arts' IIgs version of *Marble Madness* a few days ago. It is, of course, copy-protected. However, I came up with the following steps to make it bypass the copy-protection scheme:

```
CALL-151
RENAME DOS8.SYSTEM,DOS8.SYS
BLOAD DOS8.SYS,TSYS
2Ø83: EA EA EA 8Ø
CREATE DOS8.SYSTEM,TSYS
BSAVE DOS8.SYSTEM,TSYS,A$2ØØØ,L7Ø54
```

This in effect nullifies the instruction "JSR $37ØØ", which performs the copy-protection detection.

The unprotected game, however, is not hard-disk installable due to its hard-coding of the volume name /MusicGs.

UNK, Sask, Canada

Softkey for...

The American Challenge
Mindscape

■ Requirements

- □ Apple II/64K
- □ *COPYA* or *Disk Muncher (DM)*
- □ Blank Disk
- □ Sector Editor

1 Boot your DOS 3.3 system disk.

2 Tell DOS to ignore checksum and epilog errors and use *COPYA* to copy the protected disk.

```
POKE 47426,24
RUN COPYA
```

3 Make the following sector edit to the copy you just made.

Trk	Sct	Byte(s)	From	To
$ØØ	$ØØ	$2C	4C ØØ 4A	4C DB 43

That's it.

Greg Poulos

Softkey for...

Sub Mission
Mindscape

Sub Mission, by Tom Snyder Productions (distributed by Mindscape) is not copyable with a fast copier. So I loaded, *COPYA* and modified it so it would ignore some data marks. The *Sub Mission* disk was copied into a normal format.

Then I booted my modified *COPYA* version of *Sub Mission*. It seemed to boot fine, but after the screen says "Game: Approved", it would load the screen with your ship, read about three tracks and then hang, with the drive spinning. Apparently it was looking for a nibble count.

So I opened up my drive and booted it again, this time watching what the last three tracks that it read were. They were $Ø4, $Ø3, and SØ2 (if my memory serves correctly). Not possessing an incredible amount of assembly language knowledge, I looked around those tracks and put a JMP into the monitor ($FF59, or 4C 59 FF) in various places where other JMPs were.

What I discovered was that the disk seemed to read the tracks backwards, because if I put a JMP to the monitor on track $Ø4, the program would crash earlier than if I put it on track $Ø2.

But so far, I still was rather lost. So I searched the disk for hex bytes 89 CØ. These are used whenever the drive is turned on and are good bytes to search for when looking for nibble counts. I found a bunch of JSRs on track $Ø5, sector SØB. I thought that the instructions for reading tracks $Ø4, $Ø3, and $Ø2 were on track $Ø5. I was right. So I counted down three JSRs on track $Ø5, sector $ØB (where bytes 89 CØ were found) and NOPed out the third or fourth JSR, booted the disk and — no problem. It booted fine, and didn't hang.

1 Load *COPYA* from your system master.

```
LOAD COPYA
```

2 Next, enter the monitor and make some changes to the machine language part of *COPYA* so it ignores data and address epilogs, read errors, and the third byte of the data header.

```
CALL-151
B925:18 6Ø
B988:18 6Ø
BE48:18
B8FE:ØØ
3DØG
RUN
```

3 Follow the prompts to copy your original *Sub Mission* disk.

4 Get out a sector editor and make the following changes to track $Ø5, sector SØB of your copy.

Trk	Sct	Byte(s)	From	To
$Ø5	$ØB	SA6-A8	2Ø AF 5B	EA EA EA

Sub Mission is now fully deprotected.

If you're interested in changing the title pages, you can find them on tracks SØE and $18 with a hi-res picture searcher.

Jack Nissel

Softkey for...

Mickey's Space Adventure
Sierra On-Line

■ Requirements

- □ The original *Mickey's Space Adventure* disks
- □ 4 blank disks
- □ Sector editor
- □ DOS 3.3 system disk
- □ *COPYA*

The softkey in COMPUTIST #25 had several errors in it. One was corrected in COMPUTIST #27, but the other was not. Here is the softkey with all corrections made.

1 Boot your DOS 3.3 system disk and copy all 4 sides of your original disk.

```
RUN COPYA
```

2 When you are finished copying the disks, answer N to the "DO YOU WANT ANOTHER COPY? Y/N" prompt.

3 At the Applesoft prompt put disk 1, side 1 of your copy in the drive and enter:

```
NEW
BLOAD MICKEY - CODE
```

4 After the file has loaded, enter:

```
CALL-151
4Ø14:EA EA EA
BSAVE MICKEY - CODE,A$4ØØØ,L$26ØØ
```

You're done

Softkey for...

Where in Europe is Carmen Sandiego

Broderbund

Requirements

- □ The original disk
- □ A blank slave disk
- □ 2 blank disks
- □ *COPYA*
- □ *Copy II Plus*

If you are interested in finding out how I deprotected this title read the following, if not skip down to the step-by-step, at the end of this letter.

The first thing I did when I got this game was to load *Copy II Plus* and catalog it. On side 1, the head banged and then I got a VOLUME ØØ1 and the usual I/O error. On side 2, it was a ProDOS disk with the volume name /SIDE2 and one file named DATA.

I decided to try a simple approach. I booted my DOS 3.3 system disk, typed POKE 47426,24, RUN COPYA and tried to copy side 1. I was somewhat surprised to see it copy with no problems.

I tried to boot this copy but every time it tried to read track $ØØ it rebooted.

I loaded *Copy II Plus* so I could look through the sectors and decided to CATALOG the disk first. I could not beleive what I saw. When I tried to catalog side 1 of the original disk and view VOLUME ØØ1, I figured that the side was DOS 3.3, but on the copy of side 1 was ProDOS, CSA.SYSTEM, and 25 other files, each named for a letter of the alphabet, except for the letter Z.

Since *COPYA* had INITed the disk DOS 3.3, I decided to format a disk in ProDOS and copy the files from the *COPYA* copy to my ProDOS disk. I was hoping that the only protection on the original disk was what Broderbund was using to mask this side and make it look like DOS 3.3.

At the disk format level, both DOS 3.3 and ProDOS are identical. That is, there are 35 tracks ($ØØ-34) and each track has 16 sectors ($ØØ-ØF). Each address block begins with D5 AA 96 and ends with DE AA. Each data block begins with D5 AA AD and ends with DE AA. The differences are in the kind of data that is stored in each sector. (IE. DOS 3.3 stores it's disk usage map and directory on track $11, whereas ProDOS starts it's map and directory on track $ØØ.) That is why COPYA can copy both DOS 3.3 and ProDOS disks. COPYA is a whole-disk, sector-level copier. It only sees the sectors and doesn't pay attention to what kind of data is stored in which sectors........RDEXed

After copying the files, I used disk copy to copy side 2 of the original disk.

I booted this copy and I am happy to say I played the game all the way through, three times, without any problems.

I then booted the original disk. After the disk read started, the "ProDOS 1.4" screen came up. If I had done this first, instead of trying to catalog it, I would have known side 1 was ProDOS based. However, if I had known it was ProDOS based I would not have tried *COPYA* on it and I would not have deprotected it as quickly, if at all.

1 Boot your DOS 3.3 system disk and enter:

```
POKE 47426,24
RUN COPYA
```

2 Copy side 1 of your original disk to your slave disk.

3 Load *Copy II Plus* and format your first blank disk ProDOS with the VOLUME NAME A.

4 Use the COPY FILES mode to copy all of the files from your slave disk to your ProDOS disk.

5 Use the RENAME mode to change the VOLUME name on the ProDOS copy you just made from A to CARMEN.EUROPE

6 Use the COPY DISK mode to copy side 2 of your original disk to your second blank disk.

7 You're now all set to get Carmen back to jail.

Softkey for...

Dr. Ruth's Computer Game of Good Sex

Avalon Hill

Requirements

- □ The original *Dr. Ruth's Computer Game of Good Sex*
- □ 3 blank disks (1 disk to be used as a slave disk)
- □ Swap controller
- □ *Super IOB v1.5*
- □ DOS 3.3 system disk

The softkey in COMPUTIST #49 did not work for my disk. After the drive would start to read side 1 of my copy a second time I would get a DISK ERROR message. I made sure that my modified HELLO program was typed in right, and tried some changes in it, but it didn't do any good. I have had luck in the past by using the *Swap Controller* to deprotect some Avalon Hill games so I thought I would give it a try. The copy that I made would crash into the monitor when I tried to boot side 1. Deleting line 25 from the HELLO program, as the softkey in COMPUTIST #49 said to do, made the program run fine. Just to see what would happen at the Applesoft prompt, I did the POKE that line 25 did before I deleted it and it crashed into the monitor just as my copy did before I modified the HELLO program.

1 Boot your DOS 3.3 system disk.

2 INIT your blank disks HELLO and then delete the HELLO program. Label these disks 1, 2, and slave.

```
INIT HELLO
DELETE HELLO
```

3 Boot the original *Dr. Ruth* disk and after your drive reads the first 3 tracks reset into the monitor.

4 Move the RWTS to a safe location.

```
1900<B800.BFFFM
```

5 Put your slave disk in the drive and boot it.

```
C600G
```

6 At the Applesoft prompt enter:

```
BSAVE RWTS.DR.RUTH, A$1900, L$800
```

7 Install the controller listed below into *Super IOB* and copy side one of the original disk to disk number 1.

8 When side 1 is finished copying and you are at the Applesoft prompt, turn your original disk to side 2, take out disk number 1 and put in disk number 2, type RUN and press return to restart the controller to copy side 2.

9 When asked, press N, so as not to format either disk 1 or 2 while running the controller.

10 Boot disk 1 of your copy and when you get the Applesoft prompt, press RESET to stop the boot and enter the following:

```
UNLOCK HELLO
LOAD HELLO
```

11 After the HELLO program has loaded, enter the following:

```
25
SAVE HELLO
LOCK HELLO
```

Controller

```
1000 REM DR. RUTH CONTROLLER
1010 TK = 3 : ST = 0 : LT = 35 : CD = WR
1020 T1 = TK : GOSUB 490 : GØSUB360 : ONERR GOTO 550
1030 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST < DOS
     THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 : TK = TK + 1 : IF TK < LT THEN 1030
1060 GOSUB 490 : TK = T1 : ST = 0 : GOSUB 360
```

```
1070 GOSUB 430: GOSUB 100: ST = ST + 1: IF ST < DOS
     THEN 1070
1080 ST = 0: TK = TK + 1: IF BF = 0 AND TK < LT THEN
     1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "COPY^DONE" : END
10010 IF PEEK (6400) <> 162 THEN PRINT CHR$ (4)
     "BLOAD^RWTS.DR.RUTH,A$1900"
```

Checksums

1000	- $356B	1060	- $90D8
1010	- $3565	1070	- $98DD
1020	- $E1E8	1080	- $7422
1030	- $F7E9	1090	- $2DB2
1040	- $D35A	1100	- $1A55
1050	- $EB5B	10010	- $BF60

Les Minaker

Creatures of Might & Magic

The monster data in *Might and Magic* is contained on track $12, sector $07 to track $13, sector $0F. There are up to 8 monsters per sector. Unlike the character data, the monster data is stored in a standard, repeating manner. The data for the last monster on each sector, however, overflows onto the next sector. Assuming that the authors of *Might and Magic* wanted to maximize the disk usage, each monster is represented by 32 bytes. I haven't discovered what all the data is and it is unlikely that I will. However, I have learned what some of the data represents.

The information starts in exactly the same relative position for each sector. This table gives those starting positions and what some of the other locations are for. The number in parenthesis is the number of bytes used by that characteristic. This table is for all but the very first. In that sector, the data is still in the same format but the beginning of it is slightly advanced. The first monster (Flesh Eater) starts as if it was monster #5. The last monster on the disk is Lord Archer and there is no more monster data after it.

Monster#	1	2	3	4	5	6	7	8
Name ($0F)	0C	2C	4C	6C	8C	AC	CC	EC
Speed ($01)	21	41	61	81	A1	C1	E1	01
# of attacks ($01)	20	40	60	80	A0	C0	E0	00
Max damage ($01) .	1F	3F	5F	7F	9F	BF	DF	FF
Armour class ($01)	1E	3E	5E	7E	9E	BE	DE	FE

Here are the individual characteristics that I have been able to figure out so far for each monster.

Monster name	Speed	Attacks	Damage	Armour Class
12-Headed Hydra	16	12	10	10
16-Headed Hydra	12	16	12	15
5-Headed Hydra	12	5	8	7
8-Headed Hydra	13	4	8	10
Acidic Blob	8	2	8	1
Air Elemental	20	1	15	7
Algae Beast	1	1	6	
Alien	15	2	20	15
Arch Devil	14	1	100	16
Arch Druid	16	2	8	10
Arch Mage	25	2	8	12
Assasin	19	1	6	4
Basilisk	14	1	15	5
Banshee	12	1	10	10
Barbarian	15	2	12	8
Barbarian Chief	18	3	12	10
Barracuda	16	1	20	5
Battle Rat	12	1	3	3
Black Dragon	16	3	18	12
Black Knight	20	3	50	10
Blue Dragon	13	3	10	8
Caryatid Guard	15	1	10	5
Cave Giant	10	3	16	10
Cave Troll	10	3	11	7
Celestial Stag	19	3	10	14
Centaur	12	4	4	4
Chaotic Knight	16	3	15	14
Chimera	14	6	5	8
Cleric	12	1	8	5
Cockatrice	8	1	8	4
Crocodile	12	2	10	5
Cyclops	10	2	15	6
Dark Rider	14	4	50	15
Deadly Spores	10	1	1	2
Demon Dog	14	2	10	3
Demon King	35	5	50	30
Demon Lord	14	2	50	20
Diamond Golem	12	3	60	15
Dino Beetle	8	1	50	10
Dino Lizard	12	1	10	5
Dinosaur	6	1	200	10
Druid	14	2	8	4
Dung Beetle	8	1	8	6
Dust Demon	15	3	10	9
Earth Elemental	18	1	20	8
Electric Eel	15	1	8	5
Enchantress	15	24	6	6
Evil Eye	10	2	10	10
Executioner	14	2	12	8
Fire Ant	7	1	6	5
Fire Beetle	6	1	15	7
Fire Elemental	20	1	30	9
Fire Lizard	12	3	10	7
Flesh Eater	7	1	6	2
Frost Giant	12	1	24	10
Gargantu Ant	9	1	12	8
Gargoyle	12	4	4	5
Ghost	10	1	10	10
Ghoul	13	3	5	4
Giant Centipede	9	8	4	5
Giant Crab	12	2	10	9
Giant Leech	3	1	8	2
Giant Scorpion	13	3	8	7
Giant Sloth	14	4	8	5
Giant Spider	18	1	8	5
Giant Squid	14	8	6	5
Gnoll	10	1	8	5
Gnome	10	1	6	5
Goblin	10	1	6	4
Gold Dragon	16	5	20	10
Gorgon	12	1	12	8
Gray Dragon	16	3	15	8
Gray Minotaur	20	4	30	13
Great Sea Beast	30	1	100	12
Greater Demon	19	7	8	15
Greater Devil	19	4	15	12
Green Dragon	12	3	8	8
Gremlin	4	2	3	3
Griffin	14	3	8	7
Guardian Spirit	16	6	6	8
Guardsman	14	1	6	2
Hag	8	2	4	1
Harpy	15	3	5	3
High Cleric	18	3	16	14
Hill Troll	11	3	12	8
Hippocampus	18	4	10	12
Hippogriff	14	3	8	5
Invisible Thing	25	3	10	14
Killer Bees	16	10	2	10
Kirin	22	4	40	15
Kobold	6	1	4	4
Lamprey	15	1	8	2
Lava Beast	9	2	12	5
Lesser Demon	16	4	8	10
Lesser Devil	16	5	6	8
Lich	20	2	10	10
Locust Plague	17	10	1	5
Lord Archer	21	3	80	15
Mage	20	3	6	10
Magician	15	1	8	5
Man Eating Mare	14	3	8	6
Manticore	12	4	6	6
Mantis Warrior	16	4	12	8
Master Archer	18	8	16	10
Master Thief	20	2	8	12
Medusa	9	1	4	5
Militiaman	9	2	10	9
Minor Demon	16	2	8	5
Minor Devil	15	2	4	4
Minotaur	15	1	35	7
Mummy	7	2	20	7
Mutant Larva	2	1	3	
Naga	15	1	8	8
Natives	10	2	6	
Necromancer	17	2	8	7
Ogre	12	2	10	7
Ogre Chief	15	2	15	9
Orc	12	1	8	5
Orc Chieftain	15	2	12	10
Orc Leader	14	1	8	5
Panthro Mist	18	4	8	7
Paul Pead	19	1	30	10
Pegasus	20	3	8	4
Phantom	10	2	8	7
Phoenix	24	3	8	13
Pirate	17	1	20	8
Pirate Captain	18	3	20	10
Poltergiest	16	2	2	

Pyro Hydra	12	5	8	7
Rabid Jackal	15	1	2	3
Rabid Leper	11	1	3	
Rakshasha	14	3	5	14
Red Dragon	15	3	2Ø	12
Rhino Beetle	7	1	2Ø	7
Roc	14	3	5Ø	1Ø
Rotting Corpse	3	2	4	2
Sand Worm	8	1	2ØØ	7
Satyr	1Ø	2	8	5
Savage Shrew	13	3	5	3
Scorpion	2Ø	2	6Ø	12
Sea Dragon	32	4	5Ø	15
Sea Hag	12	3	6	8
Sea Serpent	2Ø	1	1ØØ	1Ø
Shadow Beast	18	1	5	3
Shark	24	2	14	6
Silver Dragon	16	4	16	8
Siren	13	2	8	8
Skeleton	9	1	6	3
Slither Beast	1Ø	1	8	4
Snake	17	1	3	2
Specter	12	1	12	8
Sphinx	18	3	1Ø	11
Sprite	2Ø	1	2	1Ø
Steel Golem	1Ø	2	25	15
Stone Giant	12	4	1Ø	1Ø
Stone Golem	6	1	4Ø	7
Storm Giant	14	2	3Ø	9
Strangling Vine	6	4	3	3
Sucubus Queen	2Ø	3	3Ø	2Ø
Swarming Wasps	17	1Ø	2	4
Swordsman	18	2	1Ø	6
Thief	16	1	8	3
Titan	3Ø	2	6Ø	13
Troglodyte	11	3	4	5
Troll	12	3	9	6
Unicorn	22	3	1Ø	8
Vampire	14	2	12	9
Vampire Bat	14	1	3	2
Volcano God	32	6	4Ø	3Ø
Warlock	16	1	8	8
Warrior	14	2	12	12
Warrior Cat	17	4	6	6
Water Elemental	14	1	5Ø	12
Water Rat	6	1	6	1
Werebear	14	4	8	8
Werephase Mummy	35	2	2Ø	2Ø
Werewolf	14	2	8	7
White Dragon	15	3	12	8
White Wolf	14	3	2	1Ø
Wicked Witch	14	2	6	4
Wight	12	1	1Ø	6
Wild Boar	14	1	12	3
Winged Beast	14	1	12Ø	12
Wizard	18	2	6	8
Wolverine	12	3	5	5
Wood Golem	5	2	15	5
Wraith	9	2	6	6
Wyvern	12	2	18	7
XX!XX!XX!XX!XX	18	1	1	2Ø
Yeti	13	2	1Ø	4
Zombie	2	1	8	2

Aaron Schoeffler

Softkey for...

Rings of Zilfin
Roadwar 2000

SSI

■ Requirements

□ *Super IOB v1.5*
□ *Roadwar 2ØØØ* and/or *Rings of Zilfin*
□ A sector editor (optional for APT's)
□ A couple blank disks

The Scheme

Being the game enthusiast I am, I had gone out and bought myself a couple of good adventure games. I sat down to play *Rings of Zilfin* first. After two hours of frustrating deaths, I decided to try my luck at deprotecting it.

After booting up my *Copy II Plus* v5.5, I snooped around using the sector editor. After examining some tracks, I found that the even tracks would read in with normal parameters, while the odd tracks would not. I tried getting rid of the checksums, but to no avail. This forced me to get my nibble copier out and get a raw dump. Looking around on track $Ø1, I found that the address prologue was changed from D5 AA 96 to D4 AA 96. Nothing else seemed to be different, so I wrote this down and tried it on my sector editor. To my suprise, it read in beautifully and I tried it on the rest of the odd tracks. Every one read in and I knew it was time to make a controller.

With the controller installed, I booted up my fresh backup and found it worked fine! Seeing this, I put it away and started playing *Roadwar 2ØØØ*. The same thing happened, my guys kept getting burnt by the sadists and never got anywhere. With a sigh, I got my *Super IOB* out and jokingly tried my *Rings of Zilfin* controller on it. My laughter subsided as it started to read and write it accordingly. With disbelief, I watched it do the entire disk. I tried to boot up the copy of *Roadwar* and again, it worked just like the original. Realizing my luck, I quickly renamed my controller and set about finding some APT's.

The Procedure

All you need to do is type in the controller at the end of this article and install it in *Super IOB*. Be sure to format your blank disk and write protect your backup of *Roadwar 2ØØØ*. The other sides of *Rings of Zilfin* are already copyable, and may I add, contain some interesting information. You may also want to delete the first two lines of the hello program and move all of the files to a Pronto-DOS disk. This would let you break out of the program whenever you want and load the files in a fraction of the time.

Controller

```
1ØØØ REM ROADWAR 2ØØØ & RINGS OF ZILFIN
1Ø1Ø TK = Ø:ST = Ø:LT = 35:CD = WR
1Ø2Ø T1 = TK: GOSUB 49Ø: POKE 474Ø5,24: POKE
     474Ø6,96: POKE 47497,24: POKE 47498,96
1Ø3Ø POKE 47445,213: IF TK / 2 < > INT (TK / 2)
     THEN POKE 47445,212
1Ø4Ø GOSUB 43Ø: GOSUB 1ØØ:ST = ST + 1: IF ST < DOS
     THEN 1Ø4Ø
1Ø5Ø IF BF THEN 1Ø8Ø
1Ø6Ø ST = Ø:TK = TK + 1:JK = 212: IF TK / 2 = INT
     (TK / 2) THEN JK = 213
1Ø7Ø POKE 47445,JK: IF TK < LT THEN 1Ø3Ø
1Ø8Ø GOSUB 49Ø:TK = T1:ST = Ø: GOSUB 23Ø
1Ø9Ø RESTORE
11ØØ POKE 474Ø5,2Ø8: POKE 474Ø6,19: POKE
     47497,2Ø8: POKE 47498,183
111Ø GOSUB 43Ø: GOSUB 1ØØ:ST = ST + 1: IF ST < DOS
     THEN 111Ø
112Ø ST = Ø:TK = TK + 1: IF BF = Ø AND TK < LT THEN
     111Ø
113Ø IF TK < LT THEN 1Ø2Ø
114Ø NORMAL : TEXT : HOME : PRINT "DONE^WITH^
     COPY" : END
```

Checksums

```
1ØØØ  - $356B      1Ø8Ø  - $8BBC
1Ø1Ø  - $3266      1Ø9Ø  - $B829
1Ø2Ø  - $2F76      11ØØ  - $A4E3
1Ø3Ø  - $659E      111Ø  - $9AE2
1Ø4Ø  - $7598      112Ø  - $4176
1Ø5Ø  - $E19B      113Ø  - $6BB7
1Ø6Ø  - $B8A7      114Ø  - $2ØB5
1Ø7Ø  - $6D24
```

A.P.T. for...

Roadwar 2000

SSI

I immediately noticed that this was not normal DOS. It used a late version of RDOS, which is a pain to sort out (there is a company that puts out a program which removes the RDOS protection), so I contented myself with finding out where my gang information was stored. I saved my character on a save disk I had just made, and turned off my game. Using my sector editor to snoop around my save disk I came across my gang's name. I wrote down the sector it was on, track $Ø3, sector $ØB, loaded up my gang stats and wrote them all down. I then went back to that sector and searched where everything was placed. I found almost everything, but discovered it wasn't enough to make a program out of. Anyway, here is the information I found on track $Ø3, sector $ØB.

Name of gang	$23-26
Food	$39-3A
Tires	$3B-3C
Fuel	$3D-3E
Ammo	$3F-4Ø
Guns	$41-42
Medical supplies	$53

Here is the information for the first car. The information for the rest of the cars follows in the same order.

Type of car	$64 (see chart)
Structure	$66-67
Manueverability	$68-69
Braking	$6A
Acceleration	$6B
Protection (L/R/F/B/T)	$72-76
Int. crew quality	$8Ø-84
Top. crew capacity	$85
Top. crew quality	$86-8A
Speed	$91-92

Chart of vehicles

Type	Byte
Motorcycle	$ØØ
Sidecar	$Ø1
Compact Conv.	$Ø2
Compact H.T.	$Ø3
Midsize Conv.	$Ø4
Midsize H.T.	$Ø5
Sports Car Conv.	$Ø6
Sports Car H.T.	$Ø7
Station Wagon	$Ø8
Limousine	$Ø9
Van	$ØA
Pickup Truck	$ØB
Offroad Conv.	$ØC
Offroad H.T.	$ØD
Bus	$ØE
Tractor	$ØF
Construction Veh.	$1Ø
Flatbed Truck	$11
Trailer Truck	$12

A.P.T. for...

Rings of Zilfin

SSI

I found on the second disk side that there was a file called "SA". This tiny file contained all of the information about my man. I wrote down some of my man's information and found all of it contained in different places in this file. I quickly constructed a program that loads your player file. It then asks you to confirm changes to your character. It edits every aspect of your character and saves it to disk. The place you are is also contained in this file for whom it may concern.

```
1Ø REM *
11 REM * RINGS OF ZILFIN
15 REM * EDITOR - BY
17 REM * AARON SCHOEFFLER
18 REM *
19 DC = 1:D$ = CHR$ (4): TEXT : HOME
2Ø TI$ = "RINGS^OF^ZILFIN^CHARACTER^EDITOR"
3Ø PRINT SPC( (4Ø - ( LEN (TI$))) / 2);TI$
4Ø POKE 34,2
5Ø VTAB 1Ø: PRINT "PLEASE^INSERT^SIDE^" ;:
     INVERSE : PRINT "TWO" ;: NORMAL : PRINT "^OF^
     RINGS"
6Ø PRINT "OF^ZILFIN.^PRESS^ANY^KEY^TO^
     CONTINUE=>^" ;: GET A$
65 PRINT A$
7Ø PRINT D$ "BLOAD^SA"
8Ø HOME
9Ø VTAB 5: PRINT "CHANGE^NAME?^(Y/N)^" ;: GET
     A$: IF A$ = "Y" THEN GOTO 11Ø
1ØØ GOTO 16Ø
11Ø HTAB 1: VTAB 7: PRINT "ENTER^FOUR^LETTERS^
     -^" ;:
115 NAME$ = ""
12Ø NAME$ = "" : FOR Y = Ø TO 3: HTAB 22 + Y: VTAB
     7: GET A$: PRINT A$:NAME$ = NAME$ + A$: NEXT
14Ø VTAB 1Ø: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     E$: IF E$ < > "Y" THEN GOTO 12Ø
15Ø FOR DC = 1 TO LEN (NAME$): POKE 16497 + DC,
     ASC ( MID$ (NAME$,DC,1)): NEXT
16Ø HOME : VTAB 15: PRINT "*^FATIGUE,^
     ENDURANCE,^MAG.^SKILL," : PRINT "SWRD^
     SKILL,^STRENGTH"
165 VTAB 5: PRINT "CHANGE^ALL^SKILLS^TO^9999?^
     (Y/N)^" ;: GET A$
167 PRINT A$
17Ø IF A$ < > "Y" THEN GOTO 2ØØ
18Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: IF A$ < > "Y" THEN GOTO 16Ø
19Ø FOR I = 1 TO 2: POKE 16385 + I,99: NEXT I : FOR
     I = 1 TO 2: POKE 16388 + I,99: NEXT : FOR I
     = 1 TO 2: POKE 1639Ø + I,99: NEXT : FOR I =
     1 TO 2: FOR I = 1 TO 2: POKE 16397 + I,99: NEXT
     I : FOR I = 1 TO 2: POKE 164Ø1 + I,99: NEXT I :
     FOR I = 1 TO 2: POKE 16393 + I,99: NEXT
2ØØ HOME : VTAB 5: PRINT "CHANGE^8^PLANTS^TO^
     9999?^(Y/N)" ;: GET A$: PRINT A$: IF A$ <
     > "Y" THEN GOTO 23Ø
21Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: PRINT A$: IF A$ < > "Y" THEN GOTO 2ØØ
22Ø FOR I = 1 TO 16: POKE 16411 + I,99: NEXT I
23Ø HOME : VTAB 5: PRINT "CHANGE^FOOD^TO^9999?^
     (Y/N)^" ;: GET A$: PRINT A$: IF A$ < > "Y"
     THEN GOTO 26Ø
24Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: PRINT A$: IF A$ < > "Y" THEN GOTO 23Ø
25Ø FOR I = 1 TO 2: POKE 16391 + I,99: NEXT
26Ø HOME : VTAB 5: PRINT "CHANGE^GOLD^TO^9999?^
     (Y/N)^" ;: GET A$: PRINT A$: IF A$ < > "Y"
     THEN GOTO 3ØØ
27Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: PRINT A$: IF A$ < > "Y" THEN GOTO 26Ø
28Ø FOR I = 1 TO 2: POKE 16387 + I,99: NEXT
29Ø HOME
3ØØ HOME : VTAB 5: PRINT "CHANGE^ALL^ITEMS^TO^
     99?^(Y/N)^" ;: GET A$: PRINT A$: IF A$ < >
     "Y" THEN GOTO 35Ø
31Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: PRINT A$: IF A$ < > "Y" THEN GOTO 3ØØ
32Ø FOR I = 1 TO 27: POKE 16428 + I,99: NEXT
33Ø HOME
34Ø HOME
35Ø HOME : VTAB 5: PRINT "CHANGE^BOW?^(Y/N)^"
     ;: GET A$: PRINT A$: IF A$ < > "Y" THEN
     GOTO 41Ø
36Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: PRINT A$: IF A$ < > "Y" THEN GOTO 35Ø
37Ø HOME : VTAB 7: PRINT "A^-^ASH^BOW" : PRINT
     "B^-^BROM^BOW" : PRINT : PRINT "PICK^ONE:^
     " ;: GET A$: IF A$ < > "A" AND A$ < > "B"
     THEN PRINT CHR$ (7): GOTO 37Ø
38Ø IF A$ = "A" THEN POKE 164Ø8,18
39Ø IF A$ = "B" THEN POKE 164Ø8,19
4ØØ POKE 164Ø9,99
41Ø HOME : VTAB 5: PRINT "CHANGE^SWORD?^(Y/N)^
     " ;: GET A$: PRINT A$: IF A$ < > "Y" THEN
     GOTO 5ØØ
42Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: PRINT A$: IF A$ < > "Y" THEN GOTO 41Ø
43Ø HOME : VTAB 7: PRINT "A^-^SHORT^SWORD" :
     PRINT "B^-^PELA^SWORD" : PRINT "C^-^
     SLICER" : PRINT "D^-^SLAYER" : PRINT "E^-^
     GRANDSWORD"
44Ø VTAB 13: PRINT "PICK^ONE^-^" ;: GET A$: PRINT
     A$: IF A$ < > "A" AND A$ < > "B" AND A$
     < > "C" AND A$ < > "D" AND A$ < > "E"
     THEN PRINT CHR$ (7): GOTO 43Ø
45Ø IF A$ = "A" THEN POKE 164Ø6,13
455 IF A$ = "B" THEN POKE 164Ø6,14
46Ø IF A$ = "C" THEN POKE 164Ø6,15
465 IF A$ = "D" THEN POKE 164Ø6,16
47Ø IF A$ = "E" THEN POKE 164Ø6,17
48Ø POKE 164Ø7,Ø
5ØØ HOME : VTAB 5: PRINT "CHANGE^ARMOUR?^(Y/N)^
     " ;: GET A$: PRINT A$: IF A$ < > "Y" THEN
     GOTO 6ØØ
51Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: PRINT A$: IF A$ < > "Y" THEN GOTO 5ØØ
52Ø HOME : VTAB 7: PRINT "A^-^LIGHT^ARMOUR" :
     PRINT "B^-^MEDIUM^ARMOUR" : PRINT "C^-^
     HEAVY^ARMOUR"
53Ø VTAB 11: PRINT "PICK^ONE^-^" ;: GET A$: PRINT
     A$: IF A$ < > "A" AND A$ < > "B" AND A$
     < > "C" THEN PRINT CHR$ (7): GOTO 53Ø
54Ø IF A$ = "A" THEN POKE 164Ø4,2Ø:
55Ø IF A$ = "B" THEN POKE 164Ø4,21
56Ø IF A$ = "C" THEN POKE 164Ø4,22
57Ø POKE 164Ø5,99
6ØØ HOME
61Ø VTAB 5: PRINT "CHANGE^#^OF^ARROWS^TO^9999?^
     (Y/N)^" ;: GET A$: PRINT A$: IF A$ < > "Y"
     THEN GOTO 7ØØ
62Ø VTAB 7: PRINT "ARE^YOU^SURE?^(Y/N)^" ;: GET
     A$: PRINT A$: IF A$ < > "Y" THEN GOTO 61Ø
63Ø POKE 16396,99: POKE 16397,99
```

```
700 HOME : VTAB 5: PRINT "PLEASE^RE-INSERT^SIDE^
    " ; : INVERSE : PRINT "TWO" ; : NORMAL : PRINT
    "^OF^RINGS" : PRINT "OF^ZILFIN^DISK." :
    PRINT "PRESS^ANY^KEY^TO^CONTINUE^==>^"
    ; : GET A$
705 PRINT A$
710 PRINT D$ "BSAVE^SA,A$4000,L$77"
720 TEXT : HOME : PRINT "GOODBYE" : END
```

Checksums

```
10    - $BADD      300   - $B750
11    - $E622      310   - $D2CC
15    - $15FA      320   - $B024
17    - $4D46      330   - $89C3
18    - $CC01      340   - $42E3
19    - $9A55      350   - $090A
20    - $7132      360   - $60A8
30    - $6564      370   - $66B3
40    - $C087      380   - $35E2
50    - $77FA      390   - $5CCB
60    - $2A27      400   - $6D07
65    - $E448      410   - $E446
70    - $63D2      420   - $539E
80    - $5588      430   - $4E23
90    - $0AC1      440   - $F8E4
100   - $C15C      450   - $56B4
110   - $D0AA      455   - $2914
115   - $955B      460   - $D5D1
120   - $9180      465   - $956F
140   - $ACF0      470   - $C74B
150   - $2617      480   - $35DB
160   - $0702      500   - $1097
165   - $675A      510   - $0982
167   - $0186      520   - $410D
170   - $8E99      530   - $C1C0
180   - $2148      540   - $4959
190   - $598C      550   - $1D15
200   - $0032      560   - $92BA
210   - $17DA      570   - $59C1
220   - $4A77      600   - $FF6A
230   - $1364      610   - $5C2E
240   - $0E7F      620   - $74E4
250   - $5FF3      630   - $6E1A
260   - $E7D0      700   - $82FE
270   - $38B4      705   - $0C40
280   - $E9C2      710   - $9398
290   - $9FD6      720   - $391B
```

Klaus Iden

Put your Print Shop graphics all on one data disk

For the many *Print Shop* enthusiasts who have sworn softly to themselves when the graphic they wanted could not be found by the program because you had the wrong side inserted, here's a hint that I've found quite useful.

I often use the *Print Shop* and one day wondered why Broderbund had not done the logical and put all the graphics on one side of the Graphics Library Disk. Examining the files on a disk revealed that there were a number of "show" files that served no purpose other than for advertisement (and taking up room). These could easily be eliminated. But would that leave enough room? There were 12Ø graphics on a Graphics Library Disk. Each graphic used four sectors. A DOS 3.3 disk was divided into 56Ø sectors. Tracks $ØØ-Ø2 were used by DOS and track $11 normally consisted of the catalog. Since I wanted the graphics disk for data only, DOS could be eliminated. That would gain 32 sectors for data (Tracks $Ø1 and $Ø2). Track $ØØ was not available without extensive changes to the *Print Shop* DOS, so I decided to leave it alone.

Turning to track $11, sector $ØØ, I found this was reserved for the VTOC (Volume Table Of Contents). That left me with 543 sectors for data and the catalog. Each graphic occupies four sectors and each catalog sector can list seven file names. A little arithematic showed that I should be able to get up to 131 graphics files onto one side of a disk, however I would need 19 catalog sectors rather than the 15 regularly allocated.

I recalled an article by Clay Ruth in Call -A.P.P.L.E.'s *All About DOS*. It turned out that allocating extra catalog sectors was quite simple. In the VTOC (track $11, sector $ØØ), bytes 1 and 2 pointed to the first catalog sector. Normally this was track $11, sector $ØF. If I were to change this to track $12, sector $Ø2, then I should have an extra 3 sectors for a total of (18 x 7) 126 files, more than I needed. All that required doing was to change bytes 1 and 2 of each of these new catalog sectors starting with track $12, sector $Ø2, such that they would each point to the next lower catalog sector.

Trk	Sct	Byte(s)	From	To
$12	$02	$01-02	?	12 01
$12	$01	$01-02	?	12 00
$12	$00	$01-02	?	11 0F

The final step would then be to change the VTOC to indicate that tracks $Ø1 and $Ø2 were available for data and sectors $ØØ-Ø3 of track $12 were not. Checking through my well worn *Beneath Apple DOS* manual, I came up with the appropriate changes.

Trk	Sct	Byte(s)	From	To
$11	$00	$3C	?	FF
		$3D	?	FF
		$40	?	FF
		$41	?	FF
		$81	?	F8
		$01	?	12
		$02	?	02

I booted up my sector editor and went to work. So far so good. I could CATALOG the disk, SAVE and LOAD files. Now for the real test. I transferred the graphic files using FID, and held my breath as I booted *Print Shop*. To my delight, the program was able to find all of the 12Ø graphics, although I found that I was unable to display them all by using the *Print Shop* CATALOG option. The ones that didn't show up were those on track $12. It is possible to patch the *Print Shop* program to read these extra sectors, but more on that later. If you initialize the data disk within the *Print Shop* program, you even get a message on track $ØØ, sector $ØØ that will tell you that you have a data disk that cannot be booted. My final touch was to use Beagle Bros. *Fatcat* program and organize my catalog alphabetically.

I had to use FID to transfer the files, even though it is slow and somewhat inconvenient, because I found that *Copy II Plus* did not like the unusual catalog format and would try to change it back to normal while transferring files, defeating the purpose of the exercise.

I used a sector editor to make the changes to my data disk, then thought, why not do this using the *Super IOB*. So, for those readers without access to a sector editor, here's a quick way to make your special graphics disk.

Step By Step

1 Initialize a new data disk using the *Print Shop* program.

2 Type in the controller and run *Super IOB*.

3 When asked how many drives, indicate only 1 and insert your newly initialized data disk. Use it for the original and target disk.

4 Transfer the graphics using FID or any other file mover you wish (beware of the problem with *Copy II Plus*).

Controller

```
1000 REM PRINT SHOP GRAPHICS DATA DISK MAKER
1010 TK = 17 : LT = 19 : ST = 15 : LS = 15 : CD = WR : FAST
     = 1
1020 GOSUB 490 : GOSUB 610
1030 GOSUB 499 : T1 = TK : TK = PEEK (TRK) : GOSUB
     310 : TK = T1 : GOSUB 610
1100 HOME : PRINT "FINISHED" : END
5000 DATA ^13^CHANGES
5010 DATA ^17,0.1,18
5020 DATA ^17,0.2,2
5030 DATA ^17,0,129,248,17.0,60,255
5040 DATA ^17,0,61.255,17.0,64,255
5050 DATA ^17,0,65.255
5060 DATA ^18,0,1,17,18,0.2,15
5070 DATA ^18,1.1,18,18,1.2,0
5080 DATA ^18,2.1,18,18,2.2,1
```

Checksums

1ØØØ	- $356B	5Ø2Ø	- $E7ØB
1Ø1Ø	- $6AØD	5Ø3Ø	- $5ØC3
1Ø2Ø	- $8DFC	5Ø4Ø	- $BØ92
1Ø3Ø	- $3CCB	5Ø5Ø	- $A3AØ
11ØØ	- $C526	5Ø6Ø	- $DF93
5ØØØ	- $5336	5Ø7Ø	- $7A5Ø
5Ø1Ø	- $4E4F	5Ø8Ø	- $652D

Print Shop Catalog Patch

■ **Requirements**

- □ Apple II+, IIe or IIc
- □ Modified Graphics Library Data Disks
- □ Deprotected *Print Shop* Program

The problem I had with *Print Shop* not locating the extra graphics on my modified graphics disk led to this patch that I discovered after several hours of snooping and head scratching. What had caused my original confusion was that the graphic files could be displayed under DOS with a catalog command. So why didn't *Print Shop* display them? After all, the program could load the graphics when requested.

The answer came to me late one evening after I had been searching the disk for anything that smacked of a catalog routine. Perhaps the *Print Shop* didn't use the DOS catalog routine! Perhaps the program used it's own RWTS (Read/Write/Track/Sector) routine to read in the catalog sectors. I searched the disk again, this time for 2Ø D9 Ø3 (JSR $Ø3D9), a call to the RWTS. I found several of these, one in HELLO and two in MENULIB. Examining the second call in MENULIB, I came across some interesting code starting at $8Ø32:

```
8Ø32 LDA $B7F7 Last slot accessed by DOS
8Ø35 STA $811F Store in IOB (present slot)
8Ø38 STA $812D Store in IOB (last slot)
8Ø3B LDA #$ØF  ←
8Ø3D STA $8123 Store in IOB(sector)
8Ø4Ø LDA $95F4 Second drive for data disk?
8Ø43 STA $812Ø Store in IOB (drive to use)
8Ø46 LDA #$ØØ
8Ø48 STA $8ØØ4 Counter for files found
8Ø4B LDA #$ØØ
8Ø4D STA $81C2 Counter for files displayed
8Ø5Ø LDA #$81  hi-byte of IOB address
8Ø52 LDY #$1E  lo-byte of IOB address
8Ø54 JSR $Ø3D9 RWTS call
```

The $ØF caught my eye. Sector $ØF is usually the start of the catalog sectors. I quickly located an IOB (Input/Ouput Block) at $811E. An IOB is a table of 17 parameters that are used by the RWTS.

```
811E  Ø1  Table type, must be $Ø1
811F  6Ø  Slot times 16 (= 6)
812Ø  Ø1  Drive = 1
8121  ØØ  Volume number expected ($ØØ
          matches any)
8122  11  Track = 11
8123  ØØ  Sector = Ø
8124  2F  DCT (Device Characteristics
          Table) LO byte
8125  81  DCT HI byte of address = $812F
8126  ØØ  Buffer address LO byte
8127  82  Buffer address HI byte = $82ØØ
8128  ØØ  Not used
8129  ØØ  Byte count ($ØØ = 256 bytes or 1
          sector)
812A  Ø1  Command code ($ØØ = seek, $Ø1 =
          read, $Ø2 = write, $Ø3 = format)
812B  ØØ  Error code ($ØØ = no error)
812C  ØØ  Volume number of last disk acessed
812D  6Ø  Slot of last disk acessed
812E  Ø1  Drive last accessed
```

Finally, it became obvious why extra sectors were never accessed. When doing a catalog under DOS, the VTOC is read and the first catalog sector is determined from bytes 1 and 2. *Print Shop* never looked at the VTOC on a graphics disk. It assumed use of only track $11 and starting sector $ØF. Code at $8ØD5 decremented the sector count, while a value of ØØ at $82Ø1 (sector buffer) indicated the last sector thus terminating the routine.

```
8ØD5  DEC  $8123 Decrement sector
8ØD8  LDA  $82Ø1 Load byte #1 of buffer
8ØDB  BNE  $8ØE8 if = Ø continue reading
8ØDD  LDA  $8ØØ4 Load # of files found
8ØEØ  BEQ  $8ØEB branch here if = Ø
8ØE2  LDA  $81C2 Load # of files on screen
8ØE5  BNE  $8ØF4 if = Ø wait for keypress
8ØE7  RTS        Return
8ØE8  JMP  $8Ø5Ø Continue reading sectors
```

The changes needed in order for the program to operate as I wanted it to would include an initial reading of the VTOC and use of bytes 1 and 2 of each sector to point to the next directory sector, rather than the decrementing routine at $8ØD5.

I couldn't add the additional code to the end of this file as there wasn't room. The file ends at $81F8 and the buffer starts at $82ØØ. I had to find unused memory that was not occupied by program code, graphics, or used as storage areas. I filled memory from $8ØØ to $9AA5 (HIMEM is $9AA6) with $FF's and started the HELLO program. I stopped the program with my *Wildcard* after using all parts of *Print Shop* and examined the memory for possible unused areas.

There wasn't much! However, there appeared to be some memory from $98F8 to $9AA5 that looked okay. The second problem was how to get my code there. I finally decided to alter the HELLO program and append my patch and a memory move routine to the end of it and have this routine run before any other code was executed. I've used this extensively and found it to work quite satisfactorily.

Step By Step

1 Load HELLO and make some changes.

```
BLOAD HELLO
CALL-151
Ø8ØØ:4C ØØ 1Ø
1ØØØ:A2 2Ø BD ØD 1Ø 9D FF 98
1ØØ8:CA DØ F7 4C 34 Ø8 A9 11
1Ø1Ø:8D 22 81 A9 ØØ 8D 23 81
1Ø18:A9 81 AØ 1E 2Ø D9 Ø3 BØ
1Ø2Ø:ØC AD Ø1 82 8D 22 81 AD
1Ø28:Ø2 82 8D 23 81 6Ø ØØ ØØ
BSAVE HELLO, A$8ØØ, L$82E
```

Here is what this code looks like

```
Ø8ØØ  JMP  $1ØØØ    Jump to our move routine
1ØØØ  LDX  #$2Ø     Load # bytes to move
1ØØ2  LDA  $1ØØD,X  Load data from here and
1ØØ5  STA  $98FF    store here
1ØØ8  DEX           Count down
1ØØ9  BNE  $1ØØ2    if <> Ø then loop back
1ØØB  JMP  $Ø834    Back to original program
1ØØE  LDA  #$11     Initialize
1Ø1Ø  STA  $8122    IOB
1Ø13  LDA  #$ØØ     to
1Ø15  STA  $8123    read
1Ø18  LDA  #$81     the
1Ø1A  LDY  #$1E     VTOC
1Ø1C  JSR  $Ø3D9    RWTS call
1Ø1F  BCS  $1Ø2D    Branch if error occurred
1Ø21  LDA  $82Ø1    Load track and
1Ø24  STA  $8122    store in IOB
1Ø27  LDA  $82Ø2    Load sector and
1Ø2A  STA  $8123    store in IOB
1Ø2D  RTS           Return
```

2 Now change MENULIB.

```
BLOAD MENULIB
8Ø3B:AD F4 95 8D 2Ø
8Ø4Ø:81 2Ø ØØ 99 2Ø 13 99
BSAVE MENULIB, A$6ØØØ, L$21F8
```

Here's what we did.

```
8Ø3B  LDA  $95F4 Get drive # to use
8Ø3E  STA  $812Ø Store in IOB
8Ø41  JSR  $99ØØ Go to patch and read VTOC,
                 then get sector and track
                 for first directory sector
8ØD5  JSR  $9913 Go to patch and get next
                 sector to read
```

That's it. The program will now catalog your *Print Shop Graphics* disks.

Jeff Hurlburt

The PRODUCT MONITOR

Ratings

☆☆☆☆☆	SUPERB
☆☆☆☆	EXCELLENT
☆☆☆	VERY GOOD
☆☆	GOOD
☆	FAIR
✋	POOR
✋✋	BAD
✋✋✋	DEFECTIVE

System Saver IIgs

Line conditioner/ cooler/ power organizer

$99.95, from Kensington for Apple IIgs

☆☆☆☆

Were you to take a poll of hardware types on the question: "What is the major threat to micro-computer health?" it's likely that "line glitches" or "over-heating" would account for the overwhelming majority of responses. For some time IIgs owners have met these threats with line conditioner strips and fan installations— 'workable' solutions, which left us wishing for some less cumbersome, less noisy alternative. Like, "why doesn't someone put all that cooling and AC line stuff in a nice, compact case?" Finally, someone has!; and the result is a product called "System Saver IIgs'.

Perfectly matched in color and styling to your IIgs, "System Saver" fits like a glove between computer and monitor. It actually looks like part of the computer, with the nice side-benefit of raising the monitor to a more comfortable viewing level. Two slant-mounted push-bar switches complete the match, each being illuminated by twin green LED's to mimic the IIgs power-on indicator.

The switches, one labeled "MASTER" and the other "AUX", each control two grounded power outlets mounted on back of the case. "MASTER" switches sockets labeled "Computer" and "Monitor" and also turns on the fan; "AUX" switches "Printer" and "Aux". In case you need to power a plug-in transformer (as used with some modems), Kensington will supply a short extension cord free (!), when you return the warranty card.

To squelch those nasty power line gremlins, you get a hefty LC filter (two 1" toroid inductors plus capacitor) with a metal oxide varistor ("MOV") surge suppressor connected between the AC lines. Similar MOV's run from AC 'hot' (black line) and 'neutral' (white line) to 'ground'. The devices appear to be GE V13ØL2ØA equivalent types, a good long-life choice for the typical 12Ø VAC installation. Finally, "System Saver" adds a back-panel-mounted circuit breaker to protect against " ... a particularly large power surge" as well as shorts on any of the outputs. Oddly, an internal fuse on the between-lines MOV is not mentioned in the documentation. If my circuit-tracing is accurate, its function is to interrupt current flow to the "MASTER" LED's to provide a visible indication of possible MOV failure.

Boasting a 1Ø amp rating, "System Saver" can easily handle the power-control/line-conditioning chores for your entire system. Still, while it rates a solid "good" on this score, there is room for improvement. For instance, since there is space for five sockets on the back panel, there should be five; and protection during non-use would be better if both AC lines were switched, instead of just 'hot'. As to control, I suspect most IIgs owners would prefer that "Computer" have its own switch, with "AUX" for monitor, printer, etc. and "MASTER" to enable everything. This way, a single bar-press could power-up the system; and one could turn the computer on and off without switching the monitor as well.

Like many IIgs types I've put off adding the recommended in-case fan. With just memory and stereo boards, it hasn't been necessary; and I wonder about the cooling efficiency of an approach that relies chiefly upon blowing air around inside the case while taking in relatively little air from the outside. In addition to which— let's face it— Apple's fan sounds like a mainframe blower! (Whether jamming with "Instant Music" or playing "Bard's Tale", who needs hurricane sound effects?!) "System Saver", on the other hand, uses its 3", 17 CFM muffin fan to pull air through the top of the computer, QUIETLY drawing fresh air through other vents, slot openings, etc..

In order to get some idea as to whether "System Saver" cooling makes any difference, I sampled in-case temperature change with and without the unit in place over two two-hour periods. Without forced-air cooling, in-case temperature rose about 18 degrees vis-a-vis outside temperature. With "System Saver" in place, the change was only 2 degrees. In a case packed with an internal modem, co-processor board, and other goodies, this sort of difference could be a life-saver for IC's and other components which must dissipate heat.. The lesson seems clear, if you plan to expand your IIgs, something on the order of "System Saver" cooling is a necessity.

Granted, one doesn't invest in "System Saver" just to provide cooling and de-glitching; there are less expensive alternatives. But, if you're after good line protection, quiet cooling, and a solid boost in operating convenience, all in a very classy package, then Kensington's "System Saver IIgs" is THE way to go.

Lane Mastadon VS. The Blubbermen

Comic Book

$12.ØØ Infocom

☆☆☆

■ **Requires:**

□ 64K Apple II series
□ one 5¼" drive
□ second drive recommended (copy side 2 of diskette)

Perhaps it was inevitable that, barely a year after release of the first computer novel ("Portal"), the first true computer comic book would 'hit the stands'. Unlike Accolade's "Comics", Infocom's "Infocomics" editions are picture-text read-only affairs, not games. Once the story begins, you can just sit back and watch the story unfold. At the fastest of three speed settings, this provides about an hour of 'reading'.

"What?!", you gasp, "no 'user interaction'?" Fear not. Recognizing that computer types are not well disposed to hands-off entertainment, Infocom lets you 'rewind', 'fast-forward', or pause the presentation, and insert a bookmark (do a GAME SAVE). Best of all is an option to branch the presentation at numerous key points. For example, in "Lane Mastadon vs. the Blubbermen", a "tattered bikini'-clad beauty and her kid brother become the companions of our hero on a mission to stop the blubbermen of Jupiter from conquering earth. Branching options let you stay with Lane from start to finish, follow his companions when they become separated, and/or take a look at things from the blubbermen's viewpoint. Following every branch (recommended) can easily double the presentation's duration.

Granted, any space adventure featuring curvacious cuties, evil aliens, and a stalwart hero (to say nothing of his elephant-shaped craft, the mighty "Mastadon") is almost a guaranteed winner. But story line is only part of the Infocomics secret. Employing simple animation and fractal techniques, each frame, in effect, becomes several, with panned scenery, in-out zooms, fades, etc.. Add sound effects and music, PLUS a chance to grom-up on "Lane Mastadon" #1 (!), and you have a quality, super-comic-book value.

VENDORS

Infocom 125 Cambridge Park Drive, Cambridge, MA 02140 (800-262-6868)

Kensington Microware 251 Park Avenue South, New York, NY 10010 (800-535-4242, in NY call 212-475-5200)

Marshall P. Brown

Softkey for...

Mind Prober

Human Edge Software Corporation

■ **Requirements**

□ Any copy program
□ A sector editor
□ 1 blank disk
□ An Apple II is useful

Mind Prober is billed as "innovative expert systems software" to "let you see people as they really are" and, within the limitations imposed by the speed and memory of the Apple II series of computers does a surprisingly good job.

The limitations, of any expert system, are the skill of the programmer in providing the right expertise for the system and his ability to have the program ask the right questions. There are, of course, a number of other technical factors relating to both the software and hardware but without the right questions and the right expert data base the attempt fails.

Mind Prober succeeds! It is dreadfully slow, spends forever in disk reads and only has a limited data base, but on the whole is a good example of an expert system in action.

It goes without saying that it is protected. Any standard disk copy utility will copy it, it just won't run. Being a relatively sane person, I found this unacceptable.

The softkey for *Mind Prober* is simple to implement, consisting of a disk edit of a single byte and, as is often the case, was found only after long hours of looking and a little luck.

Since the disk looked so standard, I started with *Locksmith's Fastcopy* to make my work disk.

The next step was to boot my favorite utility, *Locksmith 6.Ø*'s auto boot code tracer. For those of you who haven't used this utility, I can only say it is worth the price of the entire disk. It is as close to a full blown 65Ø2-65CØ2 emulator as I have ever seen. It won't read all protected disks but I am really impressed with what it will read and track.

Step three was to boot the *Mind Prober* work copy using the emulator and then to watch for the place where things went wrong with the boot.

After much watching I noticed a stange bit of code being executed at $6Ø6Ø. It was only called once and didn't spend much time there but it did look worth examining in more detail. I dropped out of the emulator and listed the code. It looked like mostly trash but you never know. I jumped back to the emulator and instructed it to stop when the program counter equaled $6Ø6Ø.

Rebooting the program from the emulator to start everything off right, I sat back and waited for the break and then single stepped through the code. What do you know, no more trash. The program uses conditional jumps to skip over sections of bad code. The most simple approach seemed to be a RETURN at $6Ø6Ø. That was too simple, so I followed through again, a step at a time, and found a disk read called via some code hidden at $Ø3D9 which JMPs to $B7B5. On the return to $6Ø8A, an examination is made of the results of the disk read. This seemed like a good place for a RETURN and as they say "the rest is history'.

A search of the disk found the code in a file named HUMAN.OBJ.

1 Make a copy using any standard disk copy utility.

2 Using a disk search utility search for `AØ Ø1 B1 1Ø A8 DØ Ø1` and change the `AØ` to `6Ø`.

Michael Javorka

Softkey for...

Xevious

Mindscape

■ **Requirements**

□ *Xevious* original disk
□ 2 blank disk sides
□ DOS 3.3 System Master
□ A file copier

1 Boot your DOS 3.3 system disk.

2 Tell DOS to ignore checksum and epilog errors and use *COPYA* to copy the disk.

```
POKE 47426,24
RUN COPYA
```

This procedure will copy the entire disk. But when the copy is booted it resets on itself. I INITed a disk with *ProntoDOS* (probably any fast DOS will do) and then copied all the files from the deprotected disk to this disk. It worked perfectly.

3 Boot up your fast DOS, put your blank disk in the drive, then type the following:

```
NEW
INIT HELLO
```

4 Use a file copy program to copy all the files from the deprotected copy to the fast DOS disk.

Paladin

Playing Tips for...

Donkey Kong

Atari

My thanks to John Baeuer's and Paul R. Wilson for their APT's on extra lives and a super Mario for *Donkey Kong*. I would like to add some extra info on super Mario.

As Paul Wilson stated, by pressing "2" we now have a super Mario. You can have Mario become super Mario at any level, not just the Elevator level. While on the Elevator level, have your super Mario walk on the ground to where the springing I beams are falling. If you position yourself just right, the I beams will push you down through the bottom of the screen and you will appear on the top of the screen to rescue your sweetheart. If you are pushed off the bottom of the screen and don't appear at the top, then walk to the left and you should drop in on your sweetheart and Kong.

Lewis J. Shireman

† Softkey for...

Dome Simplified Bookkeeping System

Accounting By Computer

■ **Requirements**

□ Super IOB modified to read/write every other track (COMPUTIST #53)
□ Copy program to copy DOS 3.3
□ Blank disk

This is a computerized version of the widely used Dome system. I picked up a copy at what I considered a reasonable price, until I discovered what they were up to in the system. The Dome System, like other paper systems of it's kind, is built around the idea that you will buy a new set of books each year. Well, their program is set up to make you do the same!

The disk is copy protected, using several techinques, but the real catch is that once you enter the accounting year, it can't be changed! Rather upset at this device, I set out to deprotect the disk prior to using any copy. Two of the copy protection schemes, once identified, were removed by using Bill Jetzer's Modify Super IOB to Read/Write Every Other Track in COMPUTIST #53. These were altered epilog values, and altered address markers on alternate tracks (D4 AA 96 instead of D5 AA 96). I used Super IOB with Bill's controller for Colonial Conquest/Blue Powder, Grey Smoke and it worked just fine for removing these problems.

There was at least one additional problem (probably a nibble count) that is removed by copying a unmodified DOS from the DOS 3.3 Master onto the copy. After that, the program worked fine.

Klaus Iden

Softkey for...

The Factory
The Pond
M-ss-ng L-nks
Sunburst Communications

■ Requirements

- □ Apple II Plus
- □ *Super IOB* v1.5

In recent issues, (COMPUTIST #3Ø and #39), methods for softkeying several Sunburst software programs were described. These depended on either a way to reset into the monitor or a machine with 128K. In either case, the authors captured the Sunburst RWTS and used a swap controller.

For people without a means of setting into the monitor or a IIe or IIc with only 64K memory, there is another way of deprotecting these excellent educational programs. My method uses a controller that changes the third byte of both the address and data headers as needed for each track. This is fairly simple because these bytes are changed in the same sequence as the write translation table found in normal DOS 3.3 ($BA29 - $BA68). See Beneath Apple DOS, by Don Worth and Pieter Lechner, and the two previous articles for more information on this subject.

Tracks $11 to $22 are written on half-tracks on the originals and are relocated on the softkeyed versions. This method will only work for DOS 3.3 programs so check the labels. Note: *Copy II Plus* v6.Ø will also make good backups of the originals.

1 INIT a blank disk, preferably with a fast DOS.

```
INIT LOGO
```

2 Load *Super IOB 1.5* and enter the following controller.

Controller

```
1ØØØ REM SUNBURST COMMUNICATIONS CONTROLLER
1Ø1Ø TK = 3:ST = Ø:LT = 35:CD = WR
1Ø2Ø T1 = TK: GOSUB 49Ø: POKE 47426,24: REM
     IGNORE EPILOGS
1Ø22 IF TK > 16 THEN CD = Ø: GOSUB 1ØØ: POKE BUF,
     PEEK (BUF) - 1:CD = RD:PH = TK * 2:S = 1:
     GOSUB 13Ø: REM READ HALF TRACKS AFTER TRACK
     16
1Ø3Ø GOSUB 43Ø: POKE 47466, PEEK (47657 + TK):
     POKE 47356, PEEK (47668 + TK): GOSUB 1ØØ:ST
     = ST + 1: IF ST < DOS THEN 1Ø3Ø: REM CHANGE
     ADDRESS AND DATA PROLOGS EVERY TRACK
1Ø4Ø IF BF THEN 1Ø6Ø
1Ø5Ø ST = Ø:TK = TK + 1: IF TK < LT THEN 1Ø22
1Ø6Ø GOSUB 23Ø: GOSUB 49Ø:TK = T1:ST = Ø
1Ø7Ø GOSUB 43Ø: GOSUB 1ØØ:ST = ST + 1: IF ST < DOS
     THEN 1Ø7Ø
1Ø8Ø ST = Ø:TK = TK + 1: IF BF = Ø AND TK < LT THEN
     1Ø7Ø
1Ø9Ø IF TK < LT THEN 1Ø2Ø
11ØØ HOME : PRINT "FINISHED" : END
```

Checksums

```
1ØØØ  - $356B        1Ø5Ø  - $D9Ø7
1Ø1Ø  - $3565        1Ø6Ø  - $686Ø
1Ø2Ø  - $BØ78        1Ø7Ø  - $6Ø65
1Ø22  - $Ø823        1Ø8Ø  - $CCAA
1Ø3Ø  - $ED95        1Ø9Ø  - $F5AA
1Ø4Ø  - $7992        11ØØ  - $36EF
```

MOST WANTED
Softkeys

ABM *Muse*
Agent U.S.A. *Scholastic*
Airheart *Broderbund*
Algeblaster *Davidson & Associates*
Algebra I *Intelligent Tutor*
Apple Super Pilot *?*
Artificial Intelligence *Scholastic*
Balance of Power *Mindscape*
Bandits *Sirius Software*
Bank Street Filer *Broderbund*
Bank Street School Filer *Sunburst Communications*
Barron's Computer SAT *?*
Battlegroup *SSI*
Battlezone *Atarisoft*
Brain Bank *The Observatory*
Burgertime *?*
Calendar Crafter IIgs *MECC*
Captain Goodnight *?*
Certificate Library Vol. I *?*
Championship Baseball *?*
Chuck Yeager's Advanced Flight Trainer *Electronic Arts*
Colossus IV *Firebird*
Creature Venture *Softsmith Co.*
Cross Clues *Science Research*
Cross Country Rally *Softsmith Co.*
Crossword Magic *Mindscape*
David's Midnight Magic *?*
DB Master V4.Ø *Stoneware*
Deathlord *Electronic Arts*
Dome Bookkeeping Systems *Dome Accounting*
Fay: The Masked Woman *Didatech Software*
Fay's Word Rally *Didatech Software*
Fun Bunch *Unicorn*
Galaxian *Atarisoft*
Game Show *Advanced Ideas*
Garfield Deluxe Edition *DLM*
Gemstone Healer *SSI*
GoldFinger *Mindscape*
GradeBusters 1-2-3 *Gradebusters*
Gutenburg Jr. *Micromation LTD*
Handicapping System *Sports Judge*
J & S Grade Book *J & S Software*
Jigsaw *Microfun*
Joust *Atarisoft*
Legacy of the Ancients *Electronic Arts*
Little Computer Peoples House on a Disk *Activision*
Lollipop Dragon: Cursor Control Adventures *Society for Visual Ed*
Lollipop Dragon: Plotting & Programming Adventures *Society for Visual Ed*
Lollipop Dragon: Letter & Number Key Adventures *Society for Visual Ed*
Lollipop Dragon: Function Key Adventures *Society for Visual Ed*
Magic Spells *The Learning Company*
Maxi Golf *Thunder Mountain*
Micro League Baseball *Micro-league Sports*
Microzine #25 *Scholastic*
Mr. Do *Datasoft*
Mr. Pixel's Cartoon Kit *Mindscape*
Ms. Pac-Man *Atarisoft*
Never Ending Story *Datasoft*
Odin *Odesta*
Peeping Tom *Microlab*
Pensate *Penguin*
Personal Finance Manager (PFM) *Apple Computer*
PFS File & Report IIgs *Software Publishing Corp.*
Pirates *Microprose Software*
Prime Plotter *Primesoft Corp.*
Principals Assistant Library *Mindscape*
Print Master *Unision World*
Pro-Football *Sports Judge*
Publisher *Springboard*
Puzzles & Posters *MECC*
Quiz Castle *Didatech Software*
Rescue On Fractalis *Epyx*
Ruski Duck *Softsmith Co.*
Scrabble *Electronic Arts*
Snoggle *Broderbund*
Space Eggs *Sirius*
Space Journey *Mindscape*
Stellar 7 *Penguin Software*
Success with Typing v1.2 *Scholastic Software*
Super Factory *Sunburst*
Think Tank *Living Video*
Tower of Myraglen II (IIgs) *?*
Toy Shop *Broderbund*
Ultima V *Origin Systems*
Universe *Omnitrend*

COMPUTIST back issues

To order, Use the back issue order form on page 42

56 **June 1988** ■*Features:* •Apple IIgs Secret Weapon •5 Second Fastboot into Locksmith 6.0 Fastcopy •The Product Monitor •Taking the grind out of Championship Wrestling •Making some improvements to The Nibbler *Softkeys:* •2400 AD •40' Graphics Studio •Accolade Comics •Aesop's Fables •American People •Animal Hotel •Applewriter IIe •Arcade Album #1 •Arctic Antics •Ballblazer •Bard's Tale II: The Destiny Knight •Bard's Tale IIgs •Cat'n Mouse •Championship Wrestling •Charlie Brown's 1,2,3's •Cobra Cavern •Color Me •Create With Garfield •David Winfield's Batter Up! •Destroyer •Disk Optimizer II •Dragonworld •Electronic Arts Software •En Vacances •En Ville •Fantavision gs •Fight Night •Forbidden Castle •G.I. Joe •Garfield Double Dares •General Manager •Goonies •GraphicWriter 2.0 •Gutenberg, Sr. •Hacker II •Hardball •Hardball gs •Infiltrator •James Bond 007: A View to a Kill •Keyboard Kadet •Kids on Keys •Lazer Maze •Le Demenagement •Le Francais par Ordinateur: •Leisure Suit Larry in the Land of the Lounge Lizards •Les Sports •Lion's Workshop •Microzine #21 •Microzine #22 •Milliken Word Processor •Millionair II •Multiscribe v2.0 •Multiscribe v3.0 •Multiscribe IIgs v3.01c •Paris En Metro •Pitfall II •Racter •Railroad Works •Rambo: First Blood Part II •Realm of Impossibility •Same or Different •Sea Dragon •Sea Strike •Shanghai •ShowOff •Shutterbug •Silent Service IIgs •Snoopy to the Rescue •Snoopy's Reading Machine •Snoopy's Skywriter Scrambler •Space Quest I •Space Station •Spy Hunter •Spy's Adventures in Europe •Spy's Adventures in North America •Stephen King: The Mist •Story Maker •Street Sports Basketball •Sub Battle Simulator •Super Sunday Football •Talking Text Writer •Type! •Un Repas Francais •Voodoo Island •Where in Europe is Carmen Sandiego •Winnie the Pooh •Winter Games •Winter Games gs •Wordzzzearch •World Games gs ■*APTs:* •2400 AD •Kung-Fu Master •Lady Tut •Ultima V ■*Playing Tips:* •2400 AD •Deathlord •Space Quest •Ultima IV •Ultima V •Wrath of Denethenor ***IBM***■ Softkeys: •Execu-Vision •MS Word •PC-Draw •Zork I & II

55 **May 1988** ■*Features:* •A Utility to Save the Lower 8 Pages of Memory •Bard's Tale Effects Locator •How to Capture Phantasie Screen Maps •Alternate Reality Character Editor •Updating the ProDOS Block Editor •Loading Flashcalc onto your RAMcard •A Copy-protection Scheme for ProDOS •The Product Monitor •Autoduel Car Editor *Softkeys:* •Alphabet Sequencing •Animal Alphabets and Other Things •Arctic Antics •The Boars' Store •The Boars Tell Time •Career Focus •Castle Wolfenstein •Charlie Brown's 123's •Charlie Brown's ABC's •City Country Opposites •Coveted Mirror •Create With Garfield •Crypt of Medea •Customized Alphabet Drill •Customized Flash Spelling •Dig Dug •Digital Paintbrush System •Estimation •Fay: Word Hunter •Fix It •Focusing on Language Arts •Fundamental Capitalization •Fundamental Punctuation Practice •Fundamental Spelling Words in Context •The Hobbit •Homonyms in Context •Individualized Study Master •Inside Outside Shapes •Inside Outside Opposites •Leisure Suit Larry in the Land of the Lounge Lizards •Master Diagnostics II & II+ •Mastertype v2.1 •Mathematics Series •Mr. and Mrs. Potato Head •Paper Models - The Christmas Kit •Peanuts Math Matcher •Peanuts Maze Marathon •Peanuts Picture Puzzlers •Perry Mason: The Case of the Mandarin Murder •Railroad Works •Random House Library •Management Programs •Rocky's Boots v.4 •Sensible Speller •Snoopy's Reading Machine •Snoopy's Skywriter Scrambler •Snoopy to the Rescue •Snoopy Writer •Spelling Demons •Stock Market Simulation •Story Builder •Story Starter •Studio II •Test Maker •Think Quick v1.0 •Tournament Bridge •Tutorial Comprehension •Typing is a Ball, Charlie Brown •Under Fire •Word Blaster •Word Count •Word Mount •Your Personal Net Worth ■*APTs:* •Under Fire ***IBM***■ Feature: •Flight Simulator RGB Modifications

54 **April 1988** ■*Features:* •Picture Loader •How To Make DEMUFFIN PLUS •Convert Print Shop graphics into Print Master graphics •lower case letters For Your Apple II Plus •The Product Monitor •Apple IIc Paddle Fix •Softkey for Daisy Professional 'Most-Protected' Award •DOS EOR Maker *Softkeys:* A2-PB1 Pinball •Animate •Bank Street Music Writer •Boulderdash Construction Kit •California Games •Countdown to Shutdown •Coveted Mirror •Create with Garfield •Daisy Professional •Destroyer •Donkey Kong •Expedition Amazon •General Chemistry Disk #8 •Graphics Studio •Green Globs & Graphic Equations •Kalamazoo Teacher's Record Book 2.0 •Kids on Keys •Marble Madness •Math Blaster •Maxwell Manor •Peanut's Maze Marathon •Petro-Calc •Police Artist •Practical Grammar •Rendezvous •Ring Quest •Roadwar Europa •Roadwar 2000 •Rocky Horror Show •Sesame Street Electric Coloring Book Series •Sesame Street Letters for You •Sesame Street Numbers •Seven Cities of Gold •Snoopy's Reading Machine •Spy's Adventures In Europe •Spy's Demise •Super Sunday Football •Talisman •Tellstar II •Top Draw v1.01A •The American Challenge •The Dam Busters •The Science Professor •Tubeway •Vocabulary Adventure I •Winter Games •Wizards' Crown •Zero-Gravity Pinball ■*APTs:* •Expedition Amazon •Might and Magic ■*Playing Tips:* •Beauracracy •King's Quest II •Lurking Horror •Maniac Mansion •Stationfall ***IBM***■ Feature: •Introduction to IBM Disk Format, Access, and Copy-protection •Putting Sargon III on harddisk ***IBM***■ Softkeys: •Prokey 3.0 •R:base 400 •Time Manager

53 **March 1988** ■*Features:* •Modify Super IOB to read/write every other track •APT for Rings Of Zilfin: Turn yourself into a lean, mean fighting Machine •More Softkeys for M.E.C.C. software (1987) •How To Use The Electronic Art's RWTS •APT for Realms Of Darkness: Realm's Wrecker! •Putting Super Boulder Dash onto a hard disk ■*Softkeys:* •2400 A.D. •Age Of Adventure •Apple's Core II •Arcade Boot Camp •Arctic Fox •Aztec •Ballblazer •Bard's Tale IIgs •Blue Powder Gray Smoke •California Games •Championship Wrestling •Colonial Conquest •Comprehension Skills I,II •Conquering Whole Numbers •Coordinate Math •Countdown To Shutdown •Dataquest: The World Community •Destroyer •Dream House •Dream Zone •Earth Orbit Station •Equation Math •Forecast: Your At-Home Weather Station •Fraction Concepts Inc •Fraction Munchers •Fraction Practice Unlimited •GBA Championship Basketball •Genesis •GFL Football •Ghost Rider •Goonies •Grade Manager v2.3 •Great American Cross-country Road Race •Hardball IIgs •Ikari Warrior •Jenny's Journeys •Kid Niki Radical Ninja •Kung-Fu Master •Learning To Tell Time •Leisure Suit Larry •Let's Learn About Money •Let's Learn About The Library •Letters For You •Lords Of Conquest •Magic Spells •Math Blaster •Money Works •Maps & Globes: Latitude & Longitude •Marble Madness •Microzine 18,19,20,21,22,23 •Mist •Morning Star Math •Movie Monster Game •Multiplication Puzzles •Multiscribe v3.0c •Murder On The Mississippi •Music Made Easy •Mystery Sentences •Number Munchers •Numbers Count •Odell Lake •Operation Frog •Opposites Attract •Oregon Trail v1.4 •Phonics Prime Time: Blends & Digraphs •Phonics Prime Time: Vowels I, II •Puzzles & Posters •Quotient Quest •Reader Rabbit •Reading Style Inventory •Realm Of Impossibility •Sesame Street 'Crayon' series •Shanghai •Sons Of Liberty •Space Quest v2.2 •Story Book: Pixelworks •Story Tree •Subtraction Puzzles •Super Huey •Super Wordfind •Tass Times In Tonetown •Those Amazing Reading Machines III, IV •Timothy Leary's Mind Mirror •To Preserve, Protect and Defend •Tower Of Myraglen •Troll's 'MicroCoarseware' series •Webster: The Word Game •Word Munchers •Words At Work: Compound It •Words At Work: Suffix Sense •World Games •World's Greatest Baseball Game •World Karate Championship •Writer Rabbit •Zoyon Patrol ■*APTs:* •Buck Rogers •Ikari Warrior •Kung-Fu Master •Leisure Suit Larry IIgs •Marble Madness •Realm Of Darkness •Rings Of Zilfin •Space Quest IIgs •Super Boulder Dash ■*Playing Tips:* •2400 A.D. •Donkey Kong •Infiltrator •Space Quest IIgs •Spy Hunter •Swashbuckler •Thexder •Ultima II • *IBM*■ *Softkeys:* •EasyWriter 1.0, II •Zork III

52 **February 1988** ■*Features:* •The Product Monitor •Unprotecting The Unprotectable: MacIntosh Softkeys! •A.P.T. Cornucopia •APT:Alternate Reality-Dungeon: Create A Super-human •Softkey for SSI's RDOS disks: 1.ProDOS RDOS, 2.RDOS Transfer Utility •Making Cracked II Plus Disks Work On The IIc ■*Softkeys:* •Apple Gradebook v2.6 •Award Maker Plus •Black Cauldron •Black Magic •California Games •Car Builder •Color Print Shop •Computer Ambush •Concepts In Science •Disney's Comic Strip Maker •Elite •Empire I, II •European Nations & Locations •Fooblitsky •Grid Designer •H.E.R.O. •Ikari Warriors •Infiltrator II •Le Francais par Ordinateur •Little Computer People's House on a Disk ★Main Street Filer •Master Diagnostics IIe ★MegaFiler ★MegaMerge •Microzine 23 •Might & Magic ★Millionaire •Mindplay software •Music Construction Set •Nibbler •Operation Market Garden •Phantasie ★Planetfall •PrintMaster Plus •Print Shop •Questron •Regatta •Ring Quest •Ringside Seat •Rings Of Zilfin •Shanghai •Silent Service •Snooper Troops •Spy's Adventure in N. America •Super Print •Tass Times In Tonetown •Think Quick ★Transylvania •Ultima I re-release •Where in the USA is Carmen Sandiego •World Games ★Zork I (★*MacIntosh softkey*) ■*A.P.T.s:* Alternate Reality: The Dungeon •Arctic Fox •Bard's Tale II •Beyond Zork •Black Magic •Cavern Creatures •Drol •Goonies •Ikari Warriors •Zorro ■*Playing Tips:* •Beyond Castle Wolfenstein •Championship Lode Runner •Conan •King's Queen II •Lode Runner •Lurking Horror •Station Fall •Ultima IV •Zork I BM■*Softkeys:* •Lotus 1-2-3 •Flight Simulator •PFS Report I BM A.P.T.S: •Bard's Tale

COMPUTIST back issues

To order, Use the back issue order form on page 42

51 **January 1988** ■*Features:* •The Cryptarithmetic Helper •Using EDD IV to Modify Tracks And Sectors •Bard's Tale APT: Dungeon Mapper Revisited •RAMfactor mod for Laser 128 •Ultima IV APT edit-tables •The Product Monitor •Get Better Sound by using the cassette jacks •Making A Fast Boot Disk •Might & Magic APT edit-tables ■*Softkeys:* •2400 AD •Aliens •Alphabet Zoo •Amnesia •Bag Of Tricks •Bard's Tale I •Bard's Tale II •Battle Cruiser •Beach-head II •Below The Root •Black Magic •Body Awareness •Bridge 4.0 •Carriers At War •Catalyst 3.0 •Centipede •Championship Boxing •Championship Wrestling •Chessmaster 2000 •Combining The Elements •Commando •Creative Contraptions •Einstein Compiler •Fat City •Fight Night •Flight Simulator v2.0 •Fun with Direction •GBA 2-On-2 Championship Basketball •GraphicWriter v1.1RA •Growing Up Small •House-on-a-disk •Intrigue •Jet •Jungle Hunt •Kindercomp •Knowing Numbers •Kung-fu Master •Law Of The West •Learning Well series •Letters And Words •Little Computer People •Make Your Own Murder Party •Manic Mansion •Master Diagnostics •Movie Maker •Music Construction Set •Pinball Construction Set •Pitstop •Print Shop Graphics Library Holiday •Print Shop IIgs •Rendezvous •Shapes And Patterns •Silent Service •Sorcerer •Spy vs Spy I & II •Stargate •Stellar 7 •Stickybear ABCs •Stickybear Drawing •Stickybear Numbers •Stickybear Printer •Stickybear Printer Library I & II •Stickybear Townbuilder •Super Boulderdash •Temple Of Apshai Trilogy •Tomahawk •Thexder •Walt Disney's Card And Party Shop •Walt Disney's Cartoon Maker •Wings Of Fury •Word Maze •World's Greatest Baseball Game •Zork III ■*A.P.T.s:* Bard's Tale •Lode Runner •Might & Magic •Ultima IV •W. Disney's Card And Party Shop •Wizardry III •Wizardry IV ■*Playing Tips:* •Autoduel •King's Quest •Manic Mansion •Summer Games •Tass Times In Tonetown •Thexder •Where In the World is Carmen Sandiego?

50 **December 1987** ■*Features:* •Super Boulderdash APT-writer •Softkeys to Activision/ MECC/ and PFS ProDOS/ software •Double F-8 ROM space w/o motherboard surgery •Ace-Apple bimodal Switch •Using Sider hard drives 3½", 800K drives, & 5¼" drives in DOS 3.3 ■*Softkeys:* •Aliens •Alter Ego •Alternate Reality •Amazing Reading Machines •Amazon •American Challenge •Arcade Album #1 •Arithmetic Critters •Award Maker •Baseball Database •Bard's Tale II: Destiny Knight •BC's Quest for Tires •Bop & Wrestle •Champ. Boxing •Champ. Wrestling •Clock Works •Commando •Computer Prep for SAT •Conflict In Vietnam •Counting Critters •Crisis Mountain •Dataquest 50 States •Deluxe Paint II •Dino Eggs •Disney Card & Party Shop •Disney Comic Strip Maker •Draw Plus •Eidolon •Electric Crayon ABCs •Expedition Amazon •Facemaker •First Letter Fun •Fish Scales •Fun From A–Z •Game Maker •GBA Champ. Basketball •GFL Champ. Football •Graphicwriter 1.0R/1.1R •Great Road Race •Hacker II •Hardball •Infiltrator II •Instant Music •James Bond 007: A View To A Kill •Jenny's Journeys •Kung Fu Master •Little People •List Handler •Manic Mansion •Mastery Arithmetic Games •Market Place •Master of Lamp •Math Rabbit •Microzine #17 •Might and Magic •Mission In Solar System •Moebius •Music Construction Set •Music Studio •Number Munchers •Paint With Words •Paintworks Plus •Path Tactics •pfs:File •pfs:Graph •pfs:Plan •pfs:Report •pfs:Write •Phonics Prime Time •Portal •Principal's Assistant •Print Shop ProDOS 8 v1.4 •Print Shop Holiday Edition •Quickflash! •Reader Rabbit •Realm of Impossibility •Robot Odyssey I v.2.0 •Rocky Horror Show •Rocky's Boots v4.0 •Saracen •Shanghai •Silent Service •Skylab •Sound Tracks •Speedy Math •Spindizzy •Street Sports Baseball •Sub-Mission •Super Boulderdash •Tass Times in Tonetown •Thexder •Top Fuel Eliminator •Word Handler •Word Munchers •Words at Work •World Karate Champ. •Writer's Choice: Elite •Zardax v5.2.1

49 **November 1987** ■*Features:* •Eliminate some ProDOS erroneous error messages •Date/time without a clock card •Sector surgery: recover lost files •Generating Applesoft programs 'on-the-fly' •Product Monitor reviews ●*PLUS:* How to convert List Handler files into standard text files •How to make GRAPHIC.GRABBERv3 run on the IIgs •Laser 128 'absolute' RESET ■*Playing Tips:* •Bard's Tale II •Conan •Donkey Kong •Hacker I •Hard Hat Mack •Orbitron •Print Shop Companion •Spellbreaker •Spy Hunter •Ultima 4 ■*A.P.T.s:* Infiltrator •Lode Runner •Montezuma's Revenge •Swordthrust series ■*Softkeys:* •Addition Logician •Animate •Arcade Boot Camp •Arctic Fox •Bard's Tale II •Cat'n Mouse •Counting Critters •Dam Busters † •Destroyer •Draw Plus v1.0 •Dr. Ruth's Comp. Game Of Good Sex •Echo 1.0 •E.D.D. 4 •Gamemaker •Hard Ball •Infiltrator •List Handler † •Locksmith 6.0 Fastcopy † •Magic Slate •Math Critters •Millionaire •Mind Mirror •One On One •Paintworks Plus v1.0 •Paintworks Plus v1.1 •PHM Pegasus •Portal •Quotient Quest •Reader Rabbit •Saunder's Chemistry CAI •Science Toolkit •Shanghai •Strip Poker † •Super Bunny •Super Sunday •Swordthrust series † •Term Paper Writer •Thief •Top Fuel Eliminator •Typing! † •Up-n-Down •Willy Byte •Writer's Choice Elite v1.0 •Writing A Character Sketch •Writing A Narative

48 **October 1987** ■*Features:* •Dungeon Editor & Encounter Editor for Ultima III •APT for Shadowkeep •Softkey for Shadowkeep •Softkey for Apple Business Graphics ■*Softkeys:* •816 Paint GS •Amnesia •Arctic Fox •Award Maker Plus •Bard's Tale II •Betterworking Word Processor •Beyond Castle Wolfenstein •Black Magic •Bookends Extended •Bop & Wrestle •Chess 7.0 •Chessmaster 2000 •Deluxe Paint GS •Destroyer •Hacker II •Hacker II GS •Hardball •Infiltrator •Instant Music GS •J-Bird •Mabel's Mansion •Marble Madness •Mean 18 GS Golf •Megabots •Might & Magic •Miner 2049er II •Mouse Word •Music Construction Set GS •Music Studio GS •New Oregon Trail •Paintworks Plus 1.0GS •Paintworks Plus 1.0GS •Paul Whitehead Teaches Chess •PHM Pegasus •Poetry Express •Print Shop color version •Rambo: First Blood part II •Rocky Horror Show •Sargon III* •Shanghai GS •Spindizzy •TelePorter •Temple Of Apshai trilogy •Top Draw GS •Transylvania •Ultima I •World's Greatest Baseball Game

47 **September 1987** ■*Features:* •Infocom-text Reader Enhancement •Color Ultimapper mod to Ultimapper IV •Towne Mapper utility for Ultima IV •Dungeon Mapper utility for Bard's Tale ■*Hardware Corner:* Interrupting Your Apple •Softkey for Charlie Brown's 1,2,3s ■*Softkeys:* •Guitar Wizard •Gemstone Warrior •Notable Phantom •Micro Wine Companion •Stickybear Printer •Note Card Maker •Starcross •Wishbringer •Dinosaur Dig •Dam Busters •Pirate Adventure •Infiltrator •MECC software •Banner Catch •Turtle Tracks •PFS File •Microzine #12, #13, #14 •Marble Madness •Writer Rabbit •Arcticfox •Age Of Adventure •Might And Magic •Space Station •Alternate Reality •Mindshadow •Gemstone Warrior •Strip Poker •Lucifer's Realm •Manuscript Manager •Bank Street Writer III •Kids On Keys •The Missing Ring •Graphic Solution •Empire I, II •Champ. Golf

46 **August 1987** ■*Softkeys:* •Advanced Microsystems Technology programs •Word Attack •Star Blazer •Science Toolkit •The Color Enhanced Print Shop •Video Vegas •The Handlers •K.C. Deals On Wheels •Law Of The West •Break The Bank Blackjack •Foundation Course In Spanish •OGRE •Puzzles And Posters ■*Features* •The Shift Key/Lower Case Option For II+ •Amazing Computer Facts •Shape Magic utility ■*Review*: Multiscribe

45 **July 1987** ■*Softkeys:* •Mouse Calc •Sands of Egypt •Number Farm •Agent U.S.A. •Wavy Navy •Kindercomp •Flight Simulator Update •Raid over Moscow •Crime Stopper •Key Perfect 5. •The Final Conflict •Miss Mouse •Snoggle ■*Features* •Write Protecting the Microsoft RAM Card •Keys to Success on the Franklin Ace •Modified F8 ROMs on the Apple /// ■*Core* •Owner's Review of Copy Master II

44 **June 1987** ■*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 •Intern. Gran Prix ■*Features* •Making DOSless Utilities •Pixit Printer Drivers ■*Review:* Z-RAM Memory Expansion Board •Reading the Joystick

43 **May 1987** ■*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 **April 1987** ■*Softkeys:* •Light Simulator •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

COMPUTIST back issues

To order, Use the back issue order form on page 42

42 April 1987 ■*Softkeys:* •Light Simulator •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 March 1987 ■*Softkeys:* •The Periodic Table •Gemstone Warrior •Inferno •Frogger •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 February 1987 ■*Softkeys:* •Adventure Writer •E-Z Learner •Mychess II •Raster Blaster •Cranston Manor •Ghostbusters •Designer's Pencil •The American Challenge •Encyclopedia Britannica Programs •Crime Wave ■*Features* •Taking the Wiz out of Wizardry •Adding a Printer Card Driver to Newsroom ■*Core* Games of 1986

39 January 1987 ■*Softkeys:* •MIDI/8+ •Homeword v2.1 •Borrowed Time •Amazon •Speed Reader][•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

38 December 1986 ■*Softkeys:* •Cyclod •Alternate Realty •Boulder Dash I & II •Hard Hat Mack (Revisited) •The Other Side •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 November 1986 ■*Softkeys:* •Under Fire •Pegasus][•Take 1 (revisited) •Flight Simulator II v1.05 (part 2) •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 October 1986 ■*Softkeys:* •Flight Simulator II v 1.05 •AutoDuel •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 ■*APT* •Championship Lode Runner

35 September 1986 ■*Softkeys:* •Olympic Decathlon •Hi-res Cribbage •Revisiting F-15 Strike Eagle •Masquerade •The Hobbit •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 August 1986 ■*Softkeys* •Crisis Mountain •Terripin Logo •Apple Logo II •Fishies 1.0 •SpellWorks •Gumball •Rescue at Rigel •Crazey Mazey •Conan •Perry Mason: The Case of the Mandarin Murder •Koronis Rift ■*Feature:* •More ROM Running ■*Core:* •Infocom Revealed

33 July 1986 ■*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 •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 June 1986 ■*Softkeys* •Revisiting Music Construction Set •Cubit •Baudville Software •Hartley Software •Bridge •Early Games for Young Children •Tawala's Last Redoubt• •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 May 1986 ■*Softkeys* •Trivia Fever •The Original Boston Computer Diet •Lifesaver •Synergistic Software •Blazing Paddles •Zardax •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 April 1986 *Softkeys* •Millionaire •SSI's RDOS •Fantavision •Spy vs. Spy •Dragonworld •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 March 1986 *Softkeys* •Threshold •Checkers v2.1 •Microtype •Gen. & Organic Chemistry Series •Uptown Trivia •Murder by the Dozen •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 February 1986 *Softkeys* •Ultima IV •Robot Odyssey •Rendezvous •Word Attack & Classmate •Three from Mindscape •Alphabetic Keyboarding •Hacker •Disk Director •Lode Runner •MIDI/4 •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 January 1986 *Softkeys* •Microzines 1-5 •Microzines 7-9 | Microzines (alternate method) •Phi Beta Filer •Sword of Kadash •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 •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 •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 •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* •DOS-Direct Sector Access

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

22 *Softkeys* •Miner 2049er •Lode Runner •A2-PB1 Pinball •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 •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 *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 *Softkeys* •Sensible Speller for ProDOS •Sideways •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

Out-of-print back issues

To order, Use the back issue order form on page 42

Due to popular demand, these sold-out issues are available now as 'zeroxed' copies, full-sized and center stapled.

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

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

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

15 ■*Softkeys:* •Mastertype •Stickybear BOP •Tic Tac Show ■*Reader's Softkeys:* •The Financial Cookbook •Escape from Rungistan •Alien Munchies •Millionaire •Plato ■*Features:* •MREAD/MWRT Update ■*Core:* •A Boot from Drive 2 •DB Master's Data Compression Techniques ■*Whiz Kid:* •DOS and the Drive - Part One ■*Adventure Tips:* •Time Zone •Mission Asteroid •Enchanter •Zork I •Ultima • Ultima II •Death in the Caribbean •Gruds in Space •Zork III •Starcross

14 ■*Features:* •Super IOB v1.2 Update •Putting Locksmith 5.0 Fast Copy Into a Normal Binary File •Batman Decoder Ring •A fix for DiskEdit ■*Softkeys:* •Seadragon •Rocky's Boots •Knoware •PFS Software •Computer Preparation SAT •MatheMagic ■*Review:* •Boulder Dash

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

12 ■*Softkeys:* •Zoom Graphix •Flip Out •Lion's Share •Music Construction Set ■*Reader's Softkeys:* •Hi-Res Computer Golf II •Suicide •Sabatage •Millionaire •Time is Money •Type Attack ■*Features:*Pseudo-ROMs on the Franklin Ace ■*Core:* •Psychedelic Symphony •The CORE Disk Searcher •The Armonitor ■*Adventure Tips:* •Cranston Manor • Enchanter •Kabul Spy •Colossal Caves •The Witness •Pirate Adventure •Ultima III-Exodus •Adventureland

11 ■*Softkeys:* •Sensible Speller •Exodus: Ultima III ■*Readers' Softkeys:* •SoftPorn Adventure •The Einstein Compiler v5.3 •Mask of The Sun ■*Features:* •Copy II Plus v4.4C: Update Of An Old Friend •Parameter List For Essential Data Duplicator ■*Core:* •Ultimaker III •The Mapping of Ultima III •Ultima II...The Rest Of The Picture

10 ■*Softkeys:* •Arcade Machine •Bank Street Writer •Minit Man ■*Reader's Softkeys* •Senible Speller IV •EDD IV •*Krell LOGO •Canyon Climber ■*Features:* •The Controller Saver •Examining Protected Applesoft BASIC Programs •Crunchlist II ■*Core:* •Applear - Voice Aynthesis •Introducing the 65SC802 and 65SC816 Chips •Review - Dino Eggs ■*Adventure Tips:* •Cranston Manor •Zork I •Planetfall •Mission Asteroid •Time Zone •Suspended •Critical Mass •Zork II •Castle Wolfenstein

9 ■*Softkeys:* •Sensible Speller •Sierra-On-Line Software •The Visible Computer: 6502 ■*Reader's Softkeys:* •Visidex •Music Construction Set •Gold Rush •Visiterm •Cosmic Combat ■Features: •Super IOB ■*Adventure Tips:* •Pirate Adventure •Mask of the Sun •Colossal Caves •Transylvania •Death in the Caribbean •Zork II ■Core: •Word Search Generator •ProDOS to DOS •ProDOS on a Franklin Ace

8 ■*Softkeys:* •Robotron •Legacy of Llylgamyn •The Artist •Data Factory v5.0 • EDD IV ■*Reader's Softkeys:* •Spy Strikes Back •Hayden Software •Apple LOGO ■*Features:* •Review of the Bit Copiers ■*Core:* •COREfiler •ProDOS Data Encryptor ■*Adventure Tips:* •Ulysses and The Golden Fleece •Serpentine •Ultima II •Castle Wolfenstein •Death in the Caribbean •Zork I •Zork II •Gruds in Space •Enchanter •Infidel •Serpent's Star ■*Whiz Kid:* •How Data is Stored on Disk

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

6 ■*Softkeys:* •Pandora's Box •Donkey Kong •Caverns of Freitag •Visifile ■*Features:* •Program Enhancements: Quick.Bug •Personalizing A Program •Modified ROMs ■*Review* •Essential Data Duplicator •The CIA ■*Core:* •Data Bases

5 ■*Softkeys:* •Homeword •Aztec •‡Bag of Tricks •Egbert II •Starcross •Hard Hat Mack •The Home Accountant ■*Reader's Softkeys:* •Dark Crystal •Screenwriter II •Visifile •Lancaster •Bill Budge's Triolgy of Games •Sammy Lightfoot •Amper-Magic •Buzzard Bait ■*Feature:* •Getting on the Right Track

4 ■*Features:* Ultima II Character Editor ■*Softkeys:* •Ultima II •Witness •Prisoner II •Pest Patrol ■*Adventure Tips:* •Ultima II & III ■Copy II Plus Parms Update

3 ■*Softkeys:* •Bag of Tricks •Multiplan ■*Readers' Softkeys:* •Visiplot /Visitrend •Sneakers •Wizardry ■*Features* •No More Bugs: The Sequel •Hidden Locations Revealed •Map Maker ■*A.P.T.s* •Choplifter ■*Adventure Tips* •Cranston Manor •Strange Odyssey

2 ■*Softkeys:* •Magic Window II •Multiplan ■*Features:* •Parameters for Locksmith 4.1 •Page Flipper •String Plotter •Three-D Wall Draw ■*Core:* Checksums ■*Input:* •Reviews of unprotected commercial software

Send these issues and/or library disks:

Issue	Mag $4.75	Disk $9.95	Both $12.95
56	□	□	□
55	□	□	□
54	□	□	□
53	□	□	□
52	□	□	□
51	□	□	□
50	□	□	□
49	□	□	□
48	□	□	□
47	□	□	□
46	□	□	□
45	□	□	□
44	□	□	□
43	□	□	□
42	□	□	□
41	□	□	□
40	□	□	□
39	□	□	□
38	□	□	□
37	□	□	□
36	□	□	□
35	□	□	□
34	□	□	□
33	□	□	□
32	□	□	□
31	□	□	□
30	□	□	□
29	□	□	□
★28	□	□	□
27	□	□	□
26	□	□	□
25	□	□	□
24	□	□	□
★23	□	□	□
22	□	□	□
■21	○	□	○
20	□	□	□
★19	□	□	□
■18	○	□	○
■17	○	□	○
★16	□	□	□
■15	○	□	○
■14	○	□	○
■13	○	□	○
■12	○	□	○
■11	○	□	○
■10	○	□	○
■9	○	□	○
■8	○	□	○
■7	○		○
Core 3	□	□	□
■6	○	□	○
■5	○	●	●
■4	○	□	○
■3	○		○
Core 2	□		□
■2	○	□	○
1	□		□
Core 1	□		□
Best of Hardcore Computing.. **Only the disk is available**	●	□	●
Core Special Combo $10...... **(All three CORE magazines; no disks)**	□	●	●

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.

★ Limited supply; first-come-first-serve basis.

● Not available.

○ Out-of-print; only 'Zeroxed' copies for sale.

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.

What is a library disk?

A library disk is a diskette that contains programs that would normally have to be typed in 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 57. If you wish to purchase a library disk not listed on the left (under the DISK column), used the out-of-print back issues ad on page 42.

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........

Name ______________________________ ID# ________

Address ______________________________________

City ________________ State ________ Zip ________

Country ________________ Phone ________________

__________-__________-__________-__________ Exp. ________

Signature ______________________________ CP57

- US funds drawn on US bank. • Most orders 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. • Send check/money order to:

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

the COMPUTIST shopper

Software Package	II e	II gs
816 Paint	$45.00 □	$45.00 □
Airheart	$24.00 □	
Award Maker Plus	$24.00 □	
Bank Street Writer 64K	$46.00 □	
Bank Street Speller	$46.00 □	
Bank Street Mailer 64K	$46.00 □	
Bank Street Filer 64K	$46.00 □	
Bank Street Writer Plus (128K)	$46.00 □	
Bank Street Mailer 128K	$46.00 □	
Bank Street Filer 128K	$46.00 □	
Beagle Graphics	$35.00 □	
Beyond Zork	$27.00 □	
Black Cauldron	$27.00 □	
Captain Goodnight	$24.00 □	
Championship Karate	$14.00 □	
Clip Art Gallery (for Paintworks Plus)	$20.00 □	
Copy II Plus v8.0	$23.00 □	$23.00 □
Dazzle Draw	$40.00 □	
Destroyer	$25.00 □	
DiskQuirk	$18.00 □	
Draw Plus	$52.00 □	
F-15 Strike Eagle	$24.00 □	
Fantavision	$34.00 □	$40.00 □
Flight Simulator II	$36.00 □	
Gamemaker	$34.00 □	
The Graphics Studio	$42.00 □	
Hacker II	$25.00 □	$30.00 □
Hardball	$22.00 □	$28.00 □
Hitchhiker's Guide	$20.00 □	
Infiltrator II	$20.00 □	
Jet	$29.00 □	
Karateka	$24.00 □	
Knight of Diamonds	$23.00 □	
Labrinth	$20.00 □	
Legacy of Llylgamyn	$26.00 □	
Loderunner	$24.00 □	
Macroworks	$20.00 □	
Math Blaster	$28.00 □	
Math Rabbit	$27.00 □	
Mean 18, Ultimate Golf	$28.00 □	
Merlin ProDOS	$66.00 □	
Might & Magic	$32.00 □	
Millionaire II	$40.00 □	
Mousewrite	$99.00 □	$99.00 □
Multiscribe	$45.00 □	$66.00 □
Music Studio	$52.00 □	
Paintworks Plus	$52.00 □	
Portal	$30.00 □	

Software Package	II e	II gs
The Print Shop	$34.00 □	$40.00 □
Print Shop Companion	$27.00 □	
Print Shop Graphics Library:		
Disk One	$16.50 □	
Disk Two	$16.50 □	
Disk Three	$16.50 □	
Pro-Byter	$32.00 □	
Racter	$30.00 □	
Reader Rabbit	$27.00 □	$34.00 □
Rocky's Boots	$34.00 □	
Science Tool Kit: Master Module	$46.00 □	
Shanghai	$24.00 □	$30.00 □
Silent Service	$23.00 □	
Solo Flight	$23.00 □	
Space Quest	$32.00 □	$32.00 □
Star Trek- The Kobayashi Alternative	$27.00 □	
Star Trek II- The Promethean Prophecy	$27.00 □	
Stickybear Series:		
ABC's	$24.00 □	
Math I	$24.00 □	
Math II	$24.00 □	
Numbers	$24.00 □	
Reading	$24.00 □	
Shapes	$24.00 □	
Typing	$24.00 □	
Writer	$24.00 □	
Sub Mission	$27.00 □	
Summer Games	$14.00 □	
Super Macroworks	$30.00 □	
Tass Times in Tonetown	$24.00 □	$24.00 □
The Tower of Myraglen	$32.00 □	
Thexder	$23.00 □	$23.00 □
Topdraw	$66.00 □	
Type	$30.00 □	
Typing Tutor IV	$32.00 □	$32.00 □
Video Vegas	$18.00 □	
Visualizer	$53.00 □	$59.00 □
Where in USA is Carmen San Diego	$30.00 □	
Where in World is Carmen San Diego	$30.00 □	
Winter Games	$25.00 □	$27.00 □
Wizardy	$32.00 □	
Word Attack	$28.00 □	
Wordperfect w/ Spelling Checker-	$95.00 □	$95.00 □
World Games	$25.00 □	$27.00 □
Writer's Choice Elite	$60.00 □	
Writer Rabbit	$24.00 □	
Zork Trilogy	$46.00 □	

COMPUTIST is proud to offer these LOW software prices* for Apple II and IIgs software

* Prices subject to change without notice.

Because we do not 'stock' but order only the software needed, the COMPUTIST Shopper keeps operating costs to a minimum and can pass the savings on to you.

In most cases, your order is shipped within two weeks. The COMPUTIST Shopper will NOT cash your check nor charge your credit card until your software is ready to be shipped. If the software you ordered is not available, you will have the option to cancel your order, or make an alternative selection.

How To Order

- **US orders:** Check the box for your selection. For Apple IIgs software, check the box in the right-hand column.
- Please add $3 per order for shipping & handling. Orders over $200 receive **free** shipping.
- Most orders shipped UPS, so use your **street address**.
- **Washington** state, please add 7.8% sales tax.
- Offer good while supplies last.
- **Foreign Orders:** Please inquire as to appropriate shipping fees

Name____________________________ID#________

Address ________________________________

City ______________ State ________ Zip __________

Country________________ Phone________________

VISA ● ________-________-________-________ Exp. ________

Signature________________________________CP57

SoftKey Publishing PO Box 110816-T Tacoma WA 98411

● Software SUBTOTAL __________

● Plus SHIPPING (see above) __________

● WA residents Sales Tax 7.8% __________

■ TOTAL Enclosed __________

Mike Basford

Softkey for...

TK!

?

This patch will remove the copy-protection completely, as well as let TK! run as a stand alone program as was done with VISICALC.

First of all when I refer to the "B:" drive, if you have a hard disk you can substitute the appropriate drive letter for the "B:" drive. I also assume that the original "TK!" is in the "A:" drive.

1 Format one System Disk under DOS 2.0 or 2.1. Label it according to the original 'TK!' diskette.

2 Copy the (unhidden) files from the original diskette to the corresponding 2.X formatted diskette.

3 Put the original 'TK!' diskette in the "A:" drive

```
Copy con: B:SOFTARTS.(C)
```

Enter: That's all folks!
Press ^Z then ENTER

You should see one file(s) copied message. This takes care of the hidden files.

I won't tell you how to use debug or any 'patcher' programs, I assume you have a basic understanding. I assume you have DEBUG.COM on a RAM or C: or CB: drive.

4 Now for some DEBUGing.

```
DEBUG
N A:TK.COM
F CS:100 L EFFF 0
L
N B:TK2.COM
R CX
:EFFF
W
Q
B:DEBUG B:TK2.COM
E 951 90 90 90     This disables break point
E 957 90 90 90     and single step overrides
```

Note: The original TK! should now be in the "A:" drive

```
G
```

Note: Program should stop at CS:511

What we did was let TK! read all the other pieces of itself from the various disk sectors (the good and bad sectors), do all the decrypting and set up all areas.

In otherwords we let it do all the work for us. We didn't even have to fool with bad tracks, or any decrypting ourselves. The reason for writing TK2.COM with a length of "EFFF" was to reserve this program area size so when TK! ran it would build all of its routines in "our" protected program area. We can then save TK2.COM as Tk3.COM with all of TK! safely stored in our protected TK2 memory area.

```
E 511 E8
E 57C CC
G
E 57C FF
T
R
```

Note: At this point copy down all the registers and flag settings. (A shift PRTSC will do it for you.)

```
R AX 0
R BX 0
R CX EFFF
R DX 0
N B:TK3.COM
W
R AX 0
R BX B230
R CX A000
R DX 5898
```

Note: at this point "BP" register had better be zeros.

```
G
```

Press ENTER since TK! is waiting on it.

Note: TK! should now come up as it normally would.

5 Now reboot the system with a DOS system disk in A drive.

```
B:DEBUG B:TK3.COM
E 951 E8 46 FF          Restore CNTL-BREAK
                               breakpoint.
E 957 E8 51 FF
```

6 What we will do next is bypass all the sector read code/decrypt logic since TK! has already done that once and put it in our protected program area that we saved as TK3.COM.

Code that should be generated.

```
A 252 MOV SI,1230          BE3012
MOV DI,0100                BF0001
MOV AX,0                   B80000
MOV BX,B230                BB30B2
MOV CX,A000                B900A0
MOV DX,5898                BA9858
MOV SP,B2AD                BCADB2
MOV BP,0                   BD0000
JMP BX                       FFE3
N B:TK4.COM
W
Q
```

7 Place your original TK! in a safe place since we will no longer need it. Place disk with TK4.COM in "A:" drive or run from hard disk.

```
TK4
```

You should see the original copyright screen and the program will just sit there. You must press ENTER since the message "PRESS ENTER TO START" is now being bypassed. If all went well you now have an unprotected, decrypted, stand alone version of TK! solver.

Note: all of the "TK!" copy protection is removed, and you may diskcopy (or copy) and rename TK4.COM to TK.COM to anywhere in the system your little heart desires.

Other Notes:

1. Checks for specially formatted tracks are completely removed.

2. You may load all the files on the newly formatted and unprotected diskette directly to hard or RAM disk, in any sub-directory you set up.

Softkey for...

SYMPHONY ver 1.00

?

Here how to wean SYMPHONY from it's master disk craving.

```
Rename SYMPHONY.CMP SYMPHONY.XXX
DEBUG SYMPHONY.XXX
R    Find the segment where loaded and add
     1000 to it.
     DS = 0DFA -xxxx = 1DFA
     (0DFAH + 1000H = 1DFAH)
E xxxx:3A05 75    change INT 13 to INT 75H
     (see label SYMINT below)
W                            save changed file
Q                                   exit DEBUG
Rename SYMPHONY.XXX SYMPHONY.CMP
SYMPH execute this program (it modifies the
     changed interupt back to 13H since
     SYMPHONY does a checksum of itself)
; assemble, link, EXE2BIN
;
```

Execute this program before using SYMPHONY. There is no need to re-execute this program after exit from SYMPHONY, in order to use the SYMPHONY again, since it is a resident program.

To further examine the symphony for possibly other ways to solve this, using DEBUG, do:

```
DEBUG SYMPHONY.EXE
G2
T3
G8A40
T
```

At this point you will find the subroutine that fills location ds:8735 with n and int 13 (the second one) which reads the serial number from floppy disk in A. Zeroing 8735 fools Symphony, and prevents it from testing for special track/sector structure on the floppy. If location 8735 is not zeroed, debugging will not be possible after IP 8A4D, since INT 3 will be modified by Symphony.

```
SYMINT EQU      75H interupt to use
SYMFLG EQU      8735H see above for
                info
CSEG   SEGMENT
                Assume CS;CSEG,
                DS:CSEG
       ORG      100H
PROGRA:XOR      AX,AX
       MOV      ES,AX set to segment
                Ø (interupt table)
       XOR      DX,DX
       MOV      AL,SYMINT interupt
                number
       MOV      CX,4 get position in
                interupt table
       MUL      CX
       MOV      BX,AX
       TEST     WORD PTR ES:[BX],Ø
                test if set?
       JZ       DOIT no, go set it
       INT      2ØH exit to dos with
                out mod
DOIT:           get interupt routine
                address
       MOV      word ptr
                ES:[BX],offset corc
                store it at the
                appropriateint
                address
       MOV      ES:[BX+2],CS also
                store the segment
       MOV      DX,5+16 length of
                this pgm in segments
       MOV      AX,31Ø3H exit & stay
                resident
       INT      21 H this is the
                actual interupt
CORC:  CLI
       PUSH     ES save all used
                registers
       PUSH     BP
       PUSH     BX
       MOV      BP,SP
       MOV      ES,[BP+8] get
                calling segment from
                stack
       MOV      BX,[BP+6] get
                calling address from
                stack
       DEC      BX back up one
       MOV      BYTE PTR ES:[BX],13H
                store interupt 13h
                there
       MOV      BYTE PTR DS:SYMFLG,Ø
                zero out the floppy
                test flag
       POP      BX restore registers
       POP      BP
       POP      ES
       STI
       IRET
                back to
                symphony
CSEG            ENDS
```

Where are all the people who wrote and called to say that they had IBM cracks and to ask why we didn't print those as well as Apple II info.? We can only print what you send us. So take some time and write to us.

Our IBM editor has gone fishing and left a note for us to call him when we have something to edit.

.... RDEXed

Help Wanted

The IBM RDEX editor desperately needs part-time writers. Any IBM experience will do. If you've used your IBM in some unusual way or solved some problems or just found something neat to do, write to the IBM RDEX editor and let him know.

Tell your friends about the IBM RDEX. Put messages on all the bulletin boards you frequent.

Get the word out!

Send all contributions to:

IBM-RDEX editor
COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

unCLASSIFIED ADS

Mind Prober—$9, Practicalc—$9
Jingle Disk—$1, Cardware—$2
PFS:Write, File, Report—$45
Postpaid. Byron Blystone, P.O.
Box 1313, Snohomish, WA 98290.

BIBLEWORKS
The text of the entire King James
version in Appleworks W/P files.
Set of seven 3½" disks — $69.95
Texas residents add 7% sales tax
LTE, Box 777, Refugio TX 78377

Trade your unwanted software.
Send your list of programs to trade.
I have over 70 originals to trade.
Byron Blystone, P.O. Box 1313
Snohomish, WA 98290.

APPLE SALE!!
Scribe Printer $149
Modem 300 $159
80 column monitor $79
Like new with cables and manuals
(407)391-8429

Wanted! Book of Apple Software
1981-85 St. Game ect.
Frank Polosky PO Bx 9542 Pgh PA
15223

HELP: Softkey for
Bank Street Writer Plus
3 1/2" Disk
I want to install this program
on a RAM Disk. I need to
defeat the Key Disk system
Alan Zimbard, 91 Penn Rd
Scarsdale, NY 10583

Print Shop Graphics
Over 2000 Public Domain
graphics on 20 disk sides.
$16.95. PrintShop Graphics
Box 1204 Torrington, WY 82240

APPLE IIGS 3.5" disk versions of "Runins of Keledoor" and "Trivia Pyramid" are now available! You've seen my ads here before. Now get these FASTER RUNNING 3.5" versions for your APPLE IIGS - both games for $9.95! The original released APPLE IIE 5.25" versions are still available - both games only $8. Game booklets included. Send check or M.O. to Mark Whitehurst. Box 485, Franklin Park, IL 60131. All orders rushed out!

How to place an UnClassified Ad

For short ads, use this order form. For larger ads, submit typed copy with appropriate instructions. Use every character space, up to 30 characters per line, we will adjust word wrap. $5 for the first line, $1 per line after that.

Special Graphics Instructions: The first three words of the first line are printed in bold for free. If you want other words bolded, use two spaces per letter and circle the entire word. (Bold letters count as two letters.) If you want a line centered, write **CENTER** next to that line. There is no charge for centering any line.

______________________________ **$5**
______________________________ **$6**
______________________________ **$7**
______________________________ **$8**
______________________________ **$9**
______________________________ **$10**

Name ______________________ ID# ________
Address ______________________
City ____________ State ________ Zip ________
Country ____________ Phone ____________
VISA ______-______-______-______ Exp. ________
Signature ______________________ CP53

Total Number of lines: ________ $______.00
Number of bold lines________ $______.00
$Total for a single ad $______.00 *
Run my ad ______ number of times. Each run that I pay
for now will cost me only ½ the total initial cost $______.00
Final Total $______.00

*Washington orders add 7.8% sales tax. Send this form and a check-money order (funds drawn on US bank only) for the entire amount to:

COMPUTIST unCLASSIFIEDS PO Box 110846 Tacoma, WA 98411

The Hacker's Ultimate Copy & Deprotection Utility

Super IOB Collection !

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

or call: **(206) 474-5750** for VISA or MasterCard orders.

*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.

Name ______________________ ID# ________

Address ______________________________

City ____________ State ______ Zip ________

Country ____________ Phone ____________

VISA MasterCard ______-______-______-______ Exp. ________

Signature ______________________ CP57

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

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 past but the treasures within those ancient documents has been diligently transcribed to the pages of a modern reference work:

The Book(s) of Softkeys

Volume I *Compiled from issues 1-5*

contains softkeys for: •Akalabeth •Ampermagic •Apple Galaxian •Aztec •Bag of Tricks •Budge's Trilogy •Buzzard Bait •Cannonball Blitz •Casino •Data Reporter •Deadline •Disk Organizer II •Egbert II Communic. 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 listings of need-to-have programs used to make unprotected backups.

Volume II *Compiled from issues 6-10*

contains softkeys for: •Apple Cider Spider •Apple Logo •Artist Arcade Machine •Bank Street Writer •Cannonball Blitz •Canyon Climber •Caverns of Freitag •Crush, Crumble & Chomp •Data Factory V •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 4,4c,4.1c •Spy Strikes Back •Time Zone v1.1 •Visible Computer: 6502 •Visidex •Visiterm •Zaxxon ● software for: •Hayden •Sierra Online □ PLUS the ultimate cracking program: Super IOB 1.5 •••and more!

Volume III *Compiled from issues 11-15*

contains softkeys for: •Alien Addition •Alien Munchies •Alligator Mix •Comp. Prep. SAT •Cut & Paste •Demolition Division. •DLM software •EA (Electronic Arts) software •Einstein Compiler 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 1.7 •MatheMagic •Minus Mission •Millionaire •Music Construction Set •One-on-one •Penguin software •PFS software •The Quest •Rocky's Boots •Sabotage •Seadragon •Sensible Speller 4 •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.

Name ______________________________ ID# ________

Address ______________________________

City ____________ State ________ Zip ____________

Country ____________ Phone ____________

________-________-________-________ Exp. ________

Signature ______________________________ CP57

□ Volume I - $7.95
+$2 shipping/handling

□ Volume II - $12.95
+$2 shipping/handling

□ Volume III - $17.95
+$2 shipping/handling

□ All 3 volumes! - $30.00
+$2 shipping/handling

Foreign orders (except Canada and Mexico) please add $5 per book or $12 per set 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