

For The Serious User Of Apple][Computers

COMPUTIST

Issue No. 43

May 1987

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

Softkeys For:

**Graphics Expander
Information Master
Certificate Maker
Elite
Catalyst
Temple of Apshai
Spell It**

Feature:

**Capturing Locksmith
6.0 Fastcopy**

Core:

**Computer Eyes / 2 :
a review**



(Page 16)

**COMPUTIST
PO Box 110846-T
Tacoma, WA 98411**

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

Coping With COMPUTIST

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

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

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

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

PR#6

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

Control characters are specially boxed:

[6]

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

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

■ **Software Recommendations:**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
10HOME:REMCLEAR SCREEN
```

The LIST will look like:

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

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

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

```
CALL -151
```

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

When finished, return to BASIC with:

```
E003G
```

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

■ **Keying In Source Code** The source code is printed to help explain a program's operation. To key it in, you will need the S-C Assembler.

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

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

■ **CHECKSOFT Instructions:**

```
LOAD filename  
BRUNCHECKSOFT
```

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

■ **CHECKBIN Instructions:**

```
CALL -151  
BLOAD program filename
```

Install CHECKBIN at an out of the way place

```
BRUN CHECKBIN,A$6000
```

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

```
xxx.xxx [CTRL] [Y]
```

Correct the lines at which the checksums differ.



You have a LEGAL RIGHT to an unlocked backup copy

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

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

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

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

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

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

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

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

Can't Find COMPUTIST Anywhere?

Stop searching and subscribe now!

annual subscription rates (12 issues):

U.S. subscription sent third class \$32

U.S./Canada/Mexico sent First Class \$45

U.S./Canada/Mexico First Class plus library disk \$100

Other Foreign \$75

Other Foreign plus library disk \$140

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

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

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

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

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

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

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

- New Subscriber
- Please renew my current subscription
- This is a change of address (include latest address label)

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


- Other Foreign \$75
- Other Foreign plus Library Disk \$140

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

  _____ Exp. _____

Signature _____ CP43

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

Is it time to renew?

Are you moving soon?

Take time now to save time later!

40¢ per disk?

And that **INCLUDES** the Tyvek sleeve, too?

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

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

50
bulk disks
for only
\$20

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

Send your order to:
Disk Offer
SoftKey Publishing
PO Box 110846-T
Tacoma, WA 98411
Enclose \$4 shipping & handling for each set.



Please send me _____ sets (50 disks per set) of 5 1/4" SS/DD diskettes. I have enclosed \$20 PLUS \$4 shipping & handling for each set and made checks / money orders payable to SoftKey Publishing.

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

  _____ Exp. _____

Signature _____

Domestic orders only, please include street address for UPS delivery.

Washington state orders add 7.8% sales tax.

Offer good while supplies last.
Expires
May 30, 1987.

See Page 31 For Special Software Savings.

☆
**NEW
ITEM!**

\$17.95

The Book of Softkeys Volume III.

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

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

We've been working hard to compile

The Book of Softkeys Volume III.

HURRY!

Order yours now!

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

SoftKey Publishing
PO Box 110937-T
Tacoma, WA 98411

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

  _____ Exp. _____

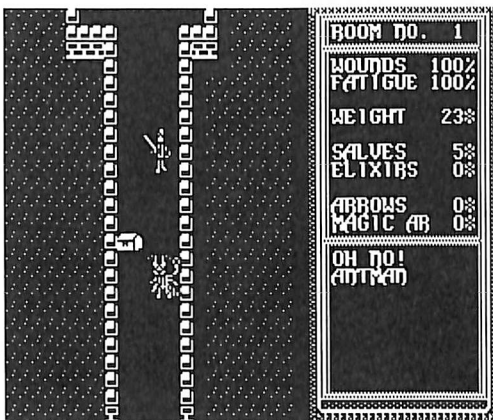
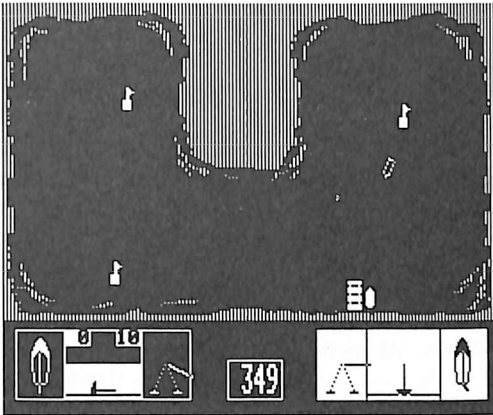
Signature _____ 42

COMPUTIST

Issue 43

May 1987

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



This month's cover:

Graphics created with Digital Visions' "Computer Eyes / 2."

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

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

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

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

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

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

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

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

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

softkeys:

15 Graphics Expander

by Jim S. Hart

22 Information Master

by Jim S. Hart

24 Certificate Maker

by Jim S. Hart

25 Elite

by Tim Scott

26 Catalyst 2.0 and 3.0

by R. Wideman

features:

10 How to Write-Protect your Slot Zero

The next best thing to having one of those old "Integer" cards is a slot zero that can be write-protected. Thus allowing you to pop into the monitor during any game! by Tim Strelchun

12 Capturing Locksmith 6.0 Fastcopy

Tired of booting your Locksmith diskette just to get to it's Fastcopy program? With this article, you can put this handy utility on the same disks with the rest of your disk utility programs. by Jerry Lewis

18 Revisiting DOS to ProDOS and Back

This article features an enhanced "Dual DOS Disk Maker" program. by Brian K. Chinn

core:

16 Computer Eyes / 2: a review

Now you too can have dazzling graphics of real life objects with the help of this hardware device from Digital Vision Inc. by Gordie Lachance

APT's and tips:

28 Sword of Kadash and Rescue Raiders

by Russell Meyer

29 Ultimaker IV

by Cracker Jack Mac

departments:

4 Input

7 Readers' Softkey & Copy Exchange

Activision's **Murder On The Mississippi** by Jeff Rivett, Epyx' **Temple of Apschai Trilogy** by Richard Yu, Troll's **Troll Associates programs** by Glen Tatum, **Spell It** by Glen Tatum, Howard W. Sams' **Regatta** by Charles Taylor, Cdex' **Cdex Training programs** by Dan Agnew, Brainpower's **Think Fast** by Tony Phalen

input

Please address letters to:

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

Include your name, address and phone number.

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

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

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

Bus Monitor Confusion

I'm very interested in building the Bus Monitor described in COMPUTIST No. 36. I have several questions concerning the project. First, will it work on a Franklin Ace 1000 and if so, would I need to make any modifications to the schematic?

Secondly, I have a question about the schematic. It appears to be contradictory in the area of the 74LS377. It says "Data low nybble (LED 5)" and to the right of that it has D1,D2,D4,D8 all followed by a (LED 6).

Lastly, the schematic shows several IC's, resistors and capacitors but does not indicate which ones need to be purchased and which are on the Apple mother board. Some sort of parts list would be very helpful here.

The potential uses for the Bus are very exciting, please keep these kinds of hardware projects coming, this subscriber really enjoys them.

Warren Vartanain
Merrimack, NH

Mr. Vartanain: We forwarded your letter to Clay Harrell, the author of the Bus Monitor article. His response follows.

Bus Monitor Clarification

Indeed this reader has found a mistake in the diagram published for the Bus Monitor in COMPUTIST No. 36 page 15. On the right side, of the middle of the diagram, the article says "Data low nybble (LED 5)" and "Data high nybble (LED 6)". It should say "Data low nybble (LED 6)" and "Data high nibble (LED 5)".

And yes, I have tested the Bus Monitor on a Franklin ACE 1000 and it works fine.

All the components shown in the diagram are necessary to build the Bus Monitor, as it is a standalone device, only using the Bus on the motherboard. None of the components shown are already on the Apple motherboard. Here is a parts list anyway:

Quant.	Description
6	Hewlett Packard 5082-7340 LED
1	74LS377
1	555
2	74LS00
1	74LS74
1	74LS07
1	8 pin socket
4	14 pin sockets
1	20 pin socket
1	1 MEG Potentometer
1	SPST push button switch (for single stepping)
1	SPST toggle switch (for normal/slow selection)
1	SPDT 2 position toggle switch (single-step/auto selection)
1	SPDT 3 position toggle switch (slow/med/fast speed select)
5	1K resistors
5	.1 microfd capacitors
1	10 microfd capacitor
1	500 pfd capacitor
1	50 finger Apple plug-compatible peripheral card
1	+5 volt, 3 amp power supply
2 ft.	30 conductor ribbon cable
1	6" by 6" perforated PC board (to assemble components on)

Clay Harrell
Detroit, MI

Quicken Note

I eagerly awaited the issue of COMPUTIST with the Quicken softkey to come out. When it did, I found out that the protection code was

not in the same place as stated by Greg Robinson. This led me to another (!) long fruitless round of trying to find the protection code. No luck. I then sent Greg Robinson a letter, asking for a listing of the protection code. Within a few days it arrived. My thanks go out to Greg for his prompt response. Dissecting the code he sent me, I finally found out where it was on my disk. Intuit changed a bit of the code along with moving it. Here is a short disassembly of MY disk's code:

```
C9 AB  CMP #SAB
D0 A8  BNE [to beginning of protection]
A9 01  LDA #S00
D0 02  BNE [2 bytes forward]
A9 00  LDA #S00-----(change the 00 to 01)
A0 00  LDY #S00
D1 04  CMP ($04),Y
48     PHA
48     PHA
A5 01  LDA $01
48     PHA
A5 00  LDA $00
48     PHA
60     RTS
```

The change to make, on my disk, is at Track \$09, Sector \$08, Byte \$61, From \$00 to \$01. If it is not there, search your copied Quicken disk for the last 9-12 bytes, then check the code immediately preceeding it to see if it closely matches the disassembly above. Again my thanks go out to Greg Robinson for his quick response & letter containing the disassembled protection code. Future short softkey submitters, remember: **always try to show a disassembly**, even a short one, so that if the protection has been moved we softkey followers can go and find it in its new location.

Jim S. Hart
Jacksonville, NC

Various Softkeys

The following are softkeys for the adventures by Mindscape. The softkey for Take 1 in previous COMPUTISTs didn't work on my version. So I'm just mentioning my softkey in case somebody didn't manage to deprotect theirs.

The following procedure works on: Voodoo Island, Rambo, First Blood Part II and Forbidden Castle.

1) Enter the monitor.

CALL -151

input

2) Tell DOS to ignore most errors.

B942:18

3) Put in your system master disk and execute the COPYA program.

RUN COPYA

4) Make the following sector edits.

TRACK	SECTOR	BYTE	FROM	TO
\$00	\$0C	\$08	4C	EA
\$00	\$0C	\$09	00	EA
\$00	\$0C	\$0A	C6	EA

That's it!!

In deprotecting Take 1 from Baudville, I found that I had to use the following method:

1) Copy the disk with most any copier.

RUN COPYA

2) Make the following sector edits.

TRACK	SECTOR	BYTE	FROM	TO
\$00	\$0F	\$08	D0	EA
\$00	\$0F	\$09	F7	EA
\$00	\$06	\$11	D0	EA
\$00	\$06	\$12	F7	EA
\$00	\$06	\$2E	D0	EA
\$00	\$06	\$2F	D5	EA

That's it!!

The Scorpion
Pirate's Harbor of Sao Paulo

Various APT's

Here is an APT for **Spy Hunter** that I've known about for a long time:

When playing **Spy Hunter**, Press "P" before being dropped off by the van and you should be fully equipped with missiles, guns, smoke screen, and oil slick!.

Note: This worked on my version, it may not work on yours.

To gain extra levels in **Wizardry**, load your characters to form a party as you would normally do. Then enter the adventure's tavern, select the character who you want to improve, insert a blank disk (initialized with DOS 3.3), and do as you normally would when entering the tavern (to gain a level, that is...).

You may enter and exit the tavern as often as you like, therefore gaining as many levels as you like. Have fun!

A. E.
Pirates Harbor of Westchester

//c ROM differences

In the softkey for **Flight Simulator COMPUTIST** No. 36, on page 24, paragraph 4; Mr. Sunshine says "move the boot ROM down to RAM (assuming you don't have a //c)." Well, I have a //c and don't know what he means. Maybe you could clear this up for me.

Lyle Marentette
Madison Hights, MI

Mr. Marentette: On an Apple][plus, //e or //gs, each card slot is allocated 256 bytes of the computer's memory for "activation code." These bytes get executed whenever the user types a PR# or an IN# (and then prints or inputs a character). On a disk drive, these 256 bytes are used to load one sector off the disk and execute it as machine code thus starting the boot up sequence. Since the Apple //c does not have slots, none of this applies to it.

*This means that for all softkey procedures that involve "Boot code Tracing" (such as the one for Flight Simulator). You will have to use a copy of the boot ROM. To get a copy of the boot ROM, you can either: 1) Go to someone who has a][Plus or a //e and type: **BSAVE BOOT ROM,ASC600,LS100** or 2) Type it in from the source code presented in the article in **COMPUTIST** No. 20 entitled "Apple][boot ROM disassembly."*

After you have the boot ROM on disk, you must BLOAD it at the proper address when following a softkey procedure. For example, because the author of the Flight Simulator softkey wants you to move the code to \$1600 (this can be deduced by the command:

1600<C600.C6FFM

in step 4) instead, you would type:

BLOAD BOOT ROM, A\$1600.

Sundog APT

I discovered the following APT by mistake while I was playing **Sundog v2.0**. The first thing I did when I received this game was try to make a backup copy of it. I write-protected both sides to avoid ruining my original disk. Since the front side writes to itself, I took the write-protect tab off the front, but I forgot to take the one off the back. When I finally found the lost town of Banville, I entered the stock-exchange building and noticed what seemed to be a

program malfunction, I left and entered the building several more times and noticed that the phase level kept going up. When I finally entered the building the tenth time, I won the game.

I hope this tip helps those people who are pulling out their hair trying to solve this fantastic adventure.

Doug Renner
Greeley, CO

Format Change Request

First off, thanks for finally putting dates on your covers. It makes organization much easier. Second, I would like to throw in a topic for discussion among your readers: The ratio of space devoted to softkeys versus core/feature articles.

While softkeys are useful if you have a need to copy the program in question, they are otherwise of limited value. I have read through a lot of them in the past just to learn more about disk access, but after a while this has diminishing returns. My main interest in my Apple is getting into the real nitty-gritty inner workings, and that is why I subscribe only to your magazine. The other Apple magazines think teaching you how to use Basic PRINT statements constitutes an "in-depth" article, which perhaps explains why so many of them have folded. **COMPUTIST**, on the other hand, has provided me with much information that can't be found elsewhere (case in point: the excellent article in **COMPUTIST** No. 36 on how to build a bus monitor--wow!). But look at **COMPUTIST** No. 32 (the anniversary issue): if you don't own (or aren't interested) in the programs listed in the table of contents, then the whole issue is kind of a waste. It's issues like this that cause people to think the only reason for **COMPUTIST**'s existence is to promote piracy. But articles like the one on the bus monitor prove them wrong! How would other readers feel about limiting softkeys to, say, one-third of each issue, and devoting the rest to core articles, question & answers, product/program reviews, etc?

A. Bruce Riley, M.D.
Portsmouth, VA

Mr. Riley: Although we have received other letters saying something to the effect of "More Softkeys!", we will keep your request in mind.

input

A Few More Softkeys

First of all, my version of Treasure Island didn't work with the softkey in COMPUTIST No. 29. So I installed the Alice In Wonderland controller from COMPUTIST No. 35 and it worked fine, although you still need to initialize a disk with Pronto-DOS and then FID the files from your Super IOB copy to your Pronto-DOS initialized disk. (For those who don't have COMPUTIST No. 35, just use the normal fast controller and make line 5000 say DATA 255,255,255).

Next softkey. The Eidolon can be cracked as follows. First write a controller to change the altered address headers to normal address headers. (I cracked this a long time ago, so I forgot what they are. Simply use Bag of Tricks to find out what they are). Next load in track \$01, sector \$0F of your Super IOB copy and look for a byte \$90 in the top (first) row. Change this to a \$B0. That's it.

I also tried the Flight Simulator [[softkey on two different versions of version 2.0 (including 1.05) and it did not work. Perhaps there was a typo in the article...

I found this little machine language routine somewhere and thought I'd pass it along to your readers. Enter the monitor and type **C050 C052 C054 C057 2001:AA 55 2000<2001.2002M 2002<2000.3FFF 34:0 N** and press return. It's especially nice in color.

You have an outstanding magazine. Keep up the excellent work.

Greg Poulos
Novi, MI

High Level of Support

This is to let people know the level of support that some companies provide for their products. Some 2 or 3 years ago I purchased a Trakstar and while I believe it is without doubt the most useful piece of equipment I have ever bought, it unfortunately died on me about 3 months ago. At the time I had 4 Apples (a 2+, a clone and 2 Apple //e's) and certainly I swapped it from machine to machine depending on the particular one I was using at the time. Anyway to cut a long story short, I wrote to Midwest Microsystems and told them what had happened and all I expected (nay - hoped for!) was to be able to use their expertise to save me doing a lot of fault finding and just have to replace the one IC that had gone down.

To my surprise, I received in the mail, not only the likely fault diagnosis (correct too as it transpired) but the replacement chip and a new socket to solder it in with. The fault was a blown chip as a result of having probably inadvertently inserting the plug backwards and sending a +12v to a +5v pin.

Even more remarkable was the fact that about a month later I received a follow-up letter to ask whether I had been able to fix the Trakstar or not and to offer extra help!

In this day and age, I could not pass up such an occurrence without bringing it to your attention as this level of support clearly deserves a *pat on the back* and my earnest congratulations.

Leigh Rowan-Kelly
West Hobart Tasmania, Australia


Another Ultima III


In COMPUTIST No. 11 on page 29, you mention that there are two versions of the Ultima III program. There may be more. As I was following the softkey procedure to get the RWTS of this program, the program continued booting when it was supposed to stop. Wondering why this was so, I started to poke around. It seems that some code (2 bytes) has been added to the Boot 1 code on the disk. This changes locations 8846 to 8848 and 844 to 846 for purposes of the softkey. When this second boot is started, unlike it is described in the softkey, the program does not stop of its own volition. Instead the program realizes that changes have been made and goes through its reboot routine. This happens very quickly after disk access has been started, so I had to press reset about 1 second after disk access started. This stopped the program and left the RWTS intact in memory. The rest of the softkey worked perfectly.

Mike Crane
Provo, UT

Which DOS is That?

I was reading through the November issue today, where on page 9, I saw B. Croome's softkey for Assembly Language Tutor. This person constantly refers to this program's RWTS as strange. When he saw that the address field header was D5 AA B5, he should have

realized that the disk was probably formatted under DOS 3.2. True, it could be an altered DOS 3.3 disk, but as it turns out, it is not. What we have here is one of those programs published during the transition period from DOS 3.2 to DOS 3.3. The only real change to this DOS is the Boot 0 code which is under DOS 3.3 so the standard disk controller can boot the disk. The protection is so simple that simply pressing  during the boot will cause it to break into basic as is stated in the article. But now comes my major gripe: why all the fuss with moving the RWTS and using a program like DEMUFFIN PLUS? It really isn't necessary. Any sector editor that can read DOS 3.2 format disks can access this disk so the solution to copying it seems simple. Use Muffin supplied on your DOS 3.3 master disk. It's really as simple as that!

As a side note for those of you interested in DOS 3.2, when this disk is booted and stopped using , the INIT command is still functional and you can initialize one of your blank disks under DOS 3.2 for further study. The only problem is that this disk won't boot because there isn't a copy of Boot 0 code on Track \$0, Sector \$0. To get around this problem, I use Dos 3+2 by Joel Buckley (published in volume 3, #3 of Nibble magazine). With this utility, I can use both DOS'es at the same time. One warning to those new to 3.2 - when a disk is formatted data must first be written to any given sector before any can be read. And the formatting takes a long time. And recovery from I/O errors takes a long time. Oh Well.

A concerned citizen

Handling the Handlers

I need help. Perhaps one of your readers has a solution. I need to be able to take the data files from LIST HANDLER and convert them so they become standard text files. (LIST HANDLER was originally from Silicon Valley, but I believe it is now handled by ALS.) If anyone can help I would very much appreciate it.

James W. Patton
Littleton, CO

readers' softkey & copy exchange

Jeff Rivett's softkey for...

Murder On The Mississippi

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

Requirements:

48K Apple or clone
Murder On The Mississippi original
A blank disk

Activision, Activision. Hmmmm... I seem to recall seeing a fairly recent article which dealt with deprotecting an Activision game. Ah yes, here it is, "Softkey for Hacker," in COMPUTIST No. 28, written by... me?

This account may seem strange, but I assure you that it is an accurate description of the thoughts that went through my mind when I first decided to crack this program. I had actually begun to depend more on COMPUTIST as a resource than on my own brain. In any case, the article was there, it seemed relevant, and after I had read it, I knew it held the answer, the final, ultimate ANSWER, to LIFE, the UNIVERSE, and... Sorry, wrong story.


The protection used on Murder is very similar to that used on Hacker. In fact, the really important parts, the routines that check for mysterious things on the disk, are identical to the Hacker routines. Some of the surrounding code is a bit different, and the name of the protection checking program and where it loads are changed.

The result of all this is that the Softkey for Hacker needs little modification to work with Murder. The protection routines are in the file BOOT, which loads at \$1000, and has a length of \$500. I determined the crucial memory locations and values in the same way as for Hacker, and then set up a bypassing routine as listed below.

Here's a listing of the changes made to BOOT:

1258-	A9 FF	LDA	#\$00
125A-	85 FC	STA	\$FC
125C-	A9 55	LDA	#\$55
125E-	4C B4 5E	JMP	\$1298

Step by Step

- 1) Copy both sides of the Murder original with COPYA or a similar copier.
- 2) Boot the copy and hit  quickly to break out of HELLO.
- 3) Type BLOAD BOOT to load the offending program.

4) Enter the monitor with a CALL -151.

5) Enter the following changes:

```
1258:A9 00 85 FC A9 55 4C 98 12
```

6) Type BSAVE BOOT, A\$1000, L\$500.

Now the sleuthing endeavours of Sir Charles Foxworth and his trusty manservant Regis can continue without the senseless hindrance of copy-protection.

"A brilliant piece of deduction, your Lordship!"

"Elementary, my dear Regis."

An afterthought: I have a feeling that the zero page location, \$FC, is not crucial, and that leaves me with the observation that the value found in the other, crucial memory location is THE SAME for both Hacker and Murder on the Mississippi. How convenient.



Richard Yu's softkey for...

Temple of Apschai Trilogy

Epyx
1043 Kiel Court
Sunnyvale, CA 94089

Requirements:

Super IOB 1.5
A blank disk
Some way of getting into the Monitor
Original T.O.A.T disk
(Temple Of Apschai Trilogy)

Temple of Apschai must be one of Epyx' favorite disks, because I have seen it redone and revised many times. It has also had two sequels, *The Upper Reaches of Apschai* and *The Curse of Ra*. Now, new and updated versions of all three have been combined onto one disk, each now enhanced incredibly on a //c or 128K //e because of double hi-res!

When my friend first purchased the disk, he wanted a backup. Since COPYA couldn't provide him one, he asked me, hoping one of my COMPUTIST magazines could help. After some searching, I found that Super IOB's Swap controller works just fine!

- 1) Boot the T.O.A.T. disk.
- 2) Interrupt the program (I used my Fingerprint card) after the disk drive stops spinning and you see the hi-res title page.
- 3) Get into the monitor if you're not already there, and move T.O.A.T's RWTS into an area that won't get trashed by a boot.

1900<B800.BFFFM

4) Boot your Super IOB disk and save T.O.A.T.'s RWTS.

C600G
BSAVE RWTS.TOAT, A\$1900, L\$800

5) Install the Swap Controller below into Super IOB and RUN. Use any formatted disk as the target disk.

Have fun!

controller

```
1000 REM TEMPLE OF APSCHAI TRILOGY
1010 TK = 0 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1020 GOSUB 360 : GOSUB 490 : GOSUB 610
1030 GOSUB 360 : GOSUB 490 : GOSUB 610 : IF
      PEEK (TRK ) = LT THEN 1050
1040 TK = PEEK (TRK ) : ST = PEEK (SCT ) : GOTO
      1020
1050 HOME : PRINT "COPY^ DONE" : END
10010 PRINT CHR$ ( 4 ) "BLOAD^ RWTS.TOAT"
      ,A$1900 "
```

controller checksums

1000	-	\$356B	1040	-	\$1078
1010	-	\$2544	1050	-	\$C0B3
1020	-	\$C809	10010	-	\$E3DE
1030	-	\$FA9D			



Glen Tatum's softkey for...

Troll Associates programs

Troll Associates

Requirements:

Troll series programs such as:
Trouble In Space or *Goofy Ghost*
A way into the monitor
FID or COPY][Plus file copy
Several blank disks

A teacher friend of mine has bragged about a software package put out by Troll Associates, and the other day I got a chance to look at it. It really is a cute set of programs, and it comes with a book and a read along cassette. The program disk is protected (of course) and is very difficult to copy. However, it is not too difficult to deprotect. Most of the DOS commands work, at least the ones we need to copy the files like CATALOG, LOAD and LIST. The Applesoft files are encrypted, but that is OK. Luckily the binary and text files are not altered. So FID will take care of most of the work. Here are the steps to follow:

readers' softkey & copy exchange

1) Boot DOS 3.3. Insert the disk you want to copy the program to and make it into a slave disk.

FP
INIT HELLO
DELETE HELLO

2) Use FID or Copy][Plus to copy all of the files except the "A" types from the original to the work disk. (Do not copy the Applesoft files because they are encrypted.)

The Applesoft files will have to be decoded and transferred to normal DOS one at a time as follows:

3) Boot the original disk, and after the program loads break into the monitor. To keep Applesoft from restarting, the run flag and error flag should be cleared.

D6:00
D8:00

4) Reconnect the funny DOS and load one of the Applesoft programs.

3D0G
LOAD filename

5) Enter the system monitor.

CALL -151

6) Examine locations \$67 and \$68 to find the starting address of the Applesoft program.

67.68

The numbers you get back are the address of the beginning of the program, with the bytes stored in reverse order. Normally it is \$801, shown as 01 08, low byte first. Write down what is shown.

Do the same with \$AF and \$B0 to find the ending address.

AF.B0

Write down the contents of these as well.

7) Now to move the program to a safe place, type:

4000<800.3800M

This copies enough memory to hold a program about 48 sectors long.

8) Boot the slave diskette, enter the monitor, and move the program back down to its normal location.

C600G
CALL -151
800<4000.7800M

9) Put the start and end addresses that you wrote down in step 5 back into their respective locations. For example, if \$67 and \$68 contained 01 08, you would type:

67:01 08

Do the same for \$AF and \$B0, the end address location.

AF:xx xx

10) Return to DOS and save the file to the slave disk.

3D0G
SAVE filename

If you are working with the file "HELLO", use the name "HELLOX" instead, until all the files have been transferred. Otherwise, the slave disk's DOS will try to RUN whatever it finds as "HELLO" and possibly ruin the file you work on next. The other files can be SAVED with their regular names.

11) Repeat steps 3-10 for the rest of the Applesoft files, and you are all set. Boot the slave diskette and type:

RENAME HELLOX, HELLO

You now have a working backup of your Troll disk.



Glen Tatum's softkey for...

Spell It!

Requirements:

Senior PROM
Blank disks

I just purchased Spell It!, a spelling tutorial program, very easy to use and entertaining as well. The parameters of Copy][Plus 6.6 would not copy the disk, but it can be deprotected with the Senior PROM. Here's how:

1) Center the PROM select switch and boot Spell It! When the program has loaded the screen "Welcome to Spell It!" Please type your first name then press the return key." After a while, the main playing field will appear. The program has now fully loaded.

2) Hit the NMI button to interrupt the program. Press #6 to move all the main 64K to the aux 64K. Press Delete to enter the monitor. Flip the PROM select switch to the normal position. Put in the utility disk, type C600G to boot, and then follow the prompts to create a self booting 64K resurrect disk, with page 1 full-screen graphics. Copy the backside of the program disk normally. That's all there is to it.

3) You will need to do something similar for the editor program, which is accessed during the boot-up by holding down the E key. You will then have three disk sides instead of the two-sided original.

If you do not have a Senior Prom yet, I would say hurry and get one. It has solved a lot of problems for me.



Charles Taylor's softkey for...

Regatta

Howard W. Sams & Co.
4300 W. 62nd St.
Indianapolis, IN 46268

Requirements:

48K Apple and a way into the monitor, or 128K Apple //e (unenhanced) with XFER.BOOT and RESTORE (COMPUTIST No. 28 p. 15)
Super IOB with Swap Controller
A blank disk
Original Regatta
A computer that will boot Regatta

Regatta is a sailboat racing simulation in color hi-res graphics. The computer yachtsmen control the sail from the keyboard and control the tiller with a joystick or set of paddles. The wind is controlled by the program. One person can play for record times, or two can race each other on their choice of four courses.

When I first received Regatta it would boot on some Apples and not on others. It ran on none of my three computers. (It would boot far enough on my enhanced //e to capture RWTS but wouldn't run.) This seemed to be carrying copy protection a little too far, so I set out to do something about it.

A nibble editor revealed that Regatta was mostly standard format, except that the epilogues had been changed from DE AA to FF FF. I copied the disk with COPYA and a DOS modification. An attempt to catalog the copy showed no files. Changing some of the FF FFs on track 0 to DE AA still produced a non-booting copy. I then looked at some of the catalog sectors (track \$11) with a sector editor and saw some resemblance to a catalog, but the catalog track was mostly garbage. This led me to the conclusion that a modified RWTS was protecting this disk and causing the garbage on the catalog track. After using Super IOB and the swap controller I had an unprotected Regatta that ran on][Plus or unenhanced //e Apples.

An enhanced //e or //c may be used for the crack, but won't run the original or deprotected Regatta.

Step by Step

1) INIT the blank disk you want to put Regatta on.

INIT HELLO

2) Insert the disk containing your COMPUTIST programs and **BLOAD XFER.BOOT**.

readers' softkey & copy exchange

3) Insert the Regatta program disk and boot it into auxiliary memory.

**PR#3
CALL 768**

4) As soon as the disk drive stops, do a Ctrl-Reset.

5) Put the disk with your COMPUTIST stuff back in the drive and type:

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

6) Move the Regatta RWTS into main memory, out of the way of DOS.

1900<B800.BFFF 

7) Boot a disk with no hello program.
C600G

8) Save the Regatta RWTS on your Super IOB disk.

BSAVE RWTS.XXX,A\$1900,L\$800

When using the Swap Controller, make sure the name of the RWTS is the same as line 10010 of the controller.

9) Load the swap controller and merge it with Super IOB 1.5.

10) Change the first instruction in line 1010 of the controller to read TK=3, instead of TK=0. We are not interested in copying Regatta's DOS.

11) Run the program, copying Regatta to your initialized disk.

The files from Regatta can now be loaded into a 256K RAM card and played without disk access or wear and tear on the disk drive.



Dan Agnew's softkey for...

Cdex Training Programs

*Cdex Corporation
5050 El Camino Real
Los Altos, CA 94022
\$59.95*

Requirements:
A blank disk
Super IOB

The Cdex Training Programs are very good programs for learning how to manage a business. Only disk #1 has been protected. The other disks may be backed-up with COPYA. When I booted it up, the familiar BASIC prompt appeared in the bottom left corner of

the screen, revealing that the DOS was semi-standard. I had no success trying to copy the disk with COPYA and some POKEs to ignore DOS errors, so I decided to try using the Swap Controller method with Super IOB. The Swap Controller method worked perfectly until I booted up the disk. It loaded in the program, but everything froze. Nothing. Just an empty screen. I was ready to give up, but I remembered something. When I booted up the original earlier and tried to CATALOG it, I got a "NO BUFFERS AVAILABLE" error. This revealed that the MAXFILES value had been changed. Typing MAXFILES3 solves the problem. I decided to give it one more shot. I put a MAXFILES1 command into the boot program of my unprotected version and it worked!

The Procedure

1) Boot up some DOS (preferably a fast DOS) and initialize a blank disk with:

INIT PASS 

2) Boot up the Cdex program and press reset when the prompt appears. Enter the monitor and move the foreign RWTS to a safe location.

**CALL -151
1900<B800.BFFFF**


3) Insert a slave disk and reboot.



C600G

4) Save the foreign RWTS to your Super IOB disk.

BSAVE RWTS.CDEX,A\$1900,L\$800

5) Using the Super IOB Swap Controller, load in "RWTS.CDEX" and copy tracks \$3-\$22 of the Cdex disk to your initialized disk.

6) On your unprotected disk, load the file "PASS"  and add the following line.

LOAD PASS 
75 PRINT D\$; "MAXFILES1"
SAVE PASS 

7) The Cdex Training Program has now been unprotected.



Tony Phalen's softkey for...

Think Fast

*Brainpower, Inc.
24009 Ventura Boulevard
Suite 250
Calabasas, CA 91302*

Requirements:

48K Apple II
Think Fast original disk
COPYA from DOS 3.3 System Master
A sector editor

Think Fast is a brain utility program from Brainpower Inc. which aids one in the analysis and development of memory skills. It uses a series of timed exercises of increasing complexity to test your memory capacity.

When this disk is booted, the familiar "J" prompt appears. This tells us that the DOS is at least semi-normal, and shouldn't be too hard to deprotect. If you reset when you see this prompt, you will be put into Applesoft. However, any command, or even a series of random characters followed by a return, will result in a program hang. If you boot normal DOS and then try to catalog the Think Fast disk, you will get an I/O error. A closer examination of the disk reveals that the data field epilogs had been changed from the normal DE AA to DD AA. So, we will use COPYA with one modification to ignore the changes.

The Crack

1) Boot DOS 3.3 and run COPYA.

2) When COPYA is loaded and waiting for input, break out of it.



3) Delete line 70 (to keep COPYA from reloading its object file) and patch DOS to ignore the end of data markers.

**70
CALL -151
B925:18 60
3D0G**

4) Now restart COPYA and copy the Think Fast disk.

RUN

5) When the copy is done, boot up your favorite sector editor and make the following edits to your copy of Think Fast.

Track	Sector	Byte	From	To
\$00	\$03	\$42	\$38	\$18
\$11	\$00	\$01	\$00	\$11

The first edit tells DOS to ignore any errors that may occur when the disk is booting. The second edit tells the program where the catalog track is.



Slot Zero

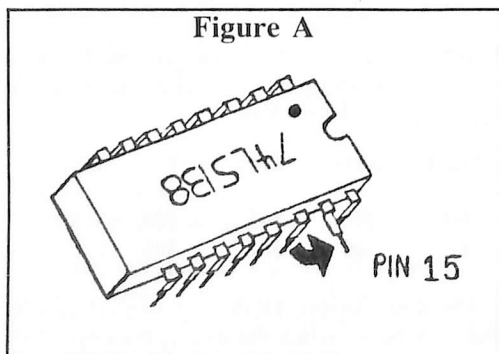
Note: COMPUTIST or Softkey Publishing cannot be held responsible for any damages that may occur while following this procedure.

by Tim Strelchun

Requirements:

Apple II or II plus w/ a 16K RAMcard
Low wattage soldering iron
Drill and bit (optional)
Three 12" pieces of insulated wire
One SPDT switch (on/on)
One 4.7K 1/4W resistor
One spare 74LS138 chip (optional)

Many programs today erase the contents of the slot zero 16K RAMcard as part of their copy protection (for instance Life Saver in COMPUTIST No. 31). This prevents us crackists from using a modified F8 ROM in our RAMcard... Well not entirely. We can always move the RAMcard to another slot, but sometimes a program will erase RAM in all the slots and it's not that enjoyable to move the RAMcard in the first place. So as an alternative to moving your 16K RAMcard, you can easily make this hardware modification (this mod only works on Apple II's and II Plus's !!!).



This modification will allow you to enable or disable the read/write signal going to slot zero (the signal sent to slot zero when you

address in the range \$C080-\$C08F) using a switch. When the read/write signal is enabled the RAMcard will act like normal. When it is disabled any addressing by a program in the range \$C080-\$C08F will have no effect (i.e. the RAMcard modes may not be changed, but the RAMcard will still operate the same). The single disadvantage to this is that in certain RAMcard modes (\$C080, \$C081, and \$C082) your computer will appear to have only 48K of RAM. The usage of this modification will be described in full, further into this article.

Making the Mod

Before doing this modification read through the procedure several times and thoroughly understand it. Also, when you solder wires to the chip be sure not to leave the soldering iron in contact with the chip pins for more than a few seconds. This precaution is necessary to prevent the chip from being overheated.

Start by turning off your computer, unplugging it, and removing the cover. Arrange the computer so that the slots are opposite you. Then look at the motherboard. Immediately in front and to the left of slot zero, at board location H2, should be a chip labeled 74LS138 (if there is not one do not do this mod!). Remove this chip from its socket and put it someplace where it will be safe, just in case you would like to remove this modification in the future.

Using your spare or your original 74LS138 chip, bend out pin 15 as shown in Figure A. Then solder one end of two of the 12" wires and one end of the 4.7K 1/4W resistor to the SPDT switch as shown in Figure B.

Now solder the wire opposite the resistor on the switch (not the center one) to pin 15 (the one you just bent out on the 74LS138). Finish by soldering the third 12" wire to the unsoldered end of the resistor and to pin 16 of the 74LS138 (when soldering to pin 16 do it near the case so that the pin will fit back in the socket). You can now insert the wire coming from the middle of the switch in the hole on

the socket where pin 15 belongs (you do not have to solder it) and then bend that wire off to the side so that when you put the chip back in the socket the wire and the bent out pin do not touch. Now put this chip back in the socket in the correct way. This whole paragraph is summarized in Figures A & B for those of you who prefer to use the right side of your brain.

If you would like this modification to be permanent choose a free area in your computer's cream colored casing (an area where the switch will not interfere with anything) and drill a hole large enough to accommodate the switch and bolt it down.

Before you try this modification out I recommend that you cover all exposed wires with tape or something insulating. Replace the cover and plug your computer into the outlet.

Testing It Out

To test the modification you just installed, try loading Integer BASIC from the system master disk. Integer BASIC should now either load or not load depending upon the original position of the switch. If it has loaded then the position the switch is in should be labeled "enable", otherwise it should be labeled "disabled."

Using the Mod

For those of you readers who do not know what the RAMcard modes are and do, here is a list describing them:

\$C080	Read from RAMcard/Write protect
\$C081	Read from ROMs/Write to RAMcard
\$C082	Read from ROMs/Write protect
\$C083	Read from RAMcard/Write to RAMcard

All the above 16K RAMcard modes operate with the switch in both positions, but in the "disable" position you cannot change the mode the RAMcard is in (when the switch is "enabled" you can change the mode by addressing the above mode location you would like to use two times if you are reading, or once

if you are writing). With this one difference you can use your RAMcard to break into protected programs without having to burn any EPROMs, utilizing the reset interrupts.

As an example of how to use this modification I will go through the steps involved in setting up the RAMcard memory, when to flip the switch, and how to use what you have set up. After this example is entered in, whenever you press the reset button, you will be dropped into the monitor (until you turn off your computer and reboot). The bold faced print following may be entered as shown from the monitor:

C081 C081

This changes the RAMcard mode to read ROM/write RAM. Whatever data that is read in the upper 16K memory area will come from the Apple ROMs. Anything addressed and attempted to be changed in the ROMs will be applied to the RAMcard. This allows you to copy a duplicate image of the ROMs into the RAMcard.

D000<D000.FFFFF

This monitor move command reads all the bytes in the range \$D000 - \$FFFF (all the ROM area that overlaps the RAMcard) and writes them to the same exact locations except onto the RAMcard. This effectively duplicates the ROM image to RAM so that you can modify it.

C083 C083

This allows you now to both read what is in the RAMcard and to change what is in the RAMcard. Notice the difference between this mode and the previous one.

FFFC:59 FF

This changes the place to which your computer jumps when a reset interrupt is generated. Your computer will now jump during a reset to \$FF59 (the command has the low-order byte first) which is the beginning of the monitor program. If a more visible change is your preference, change the monitor prompt to a colon with:

FF6A:BA

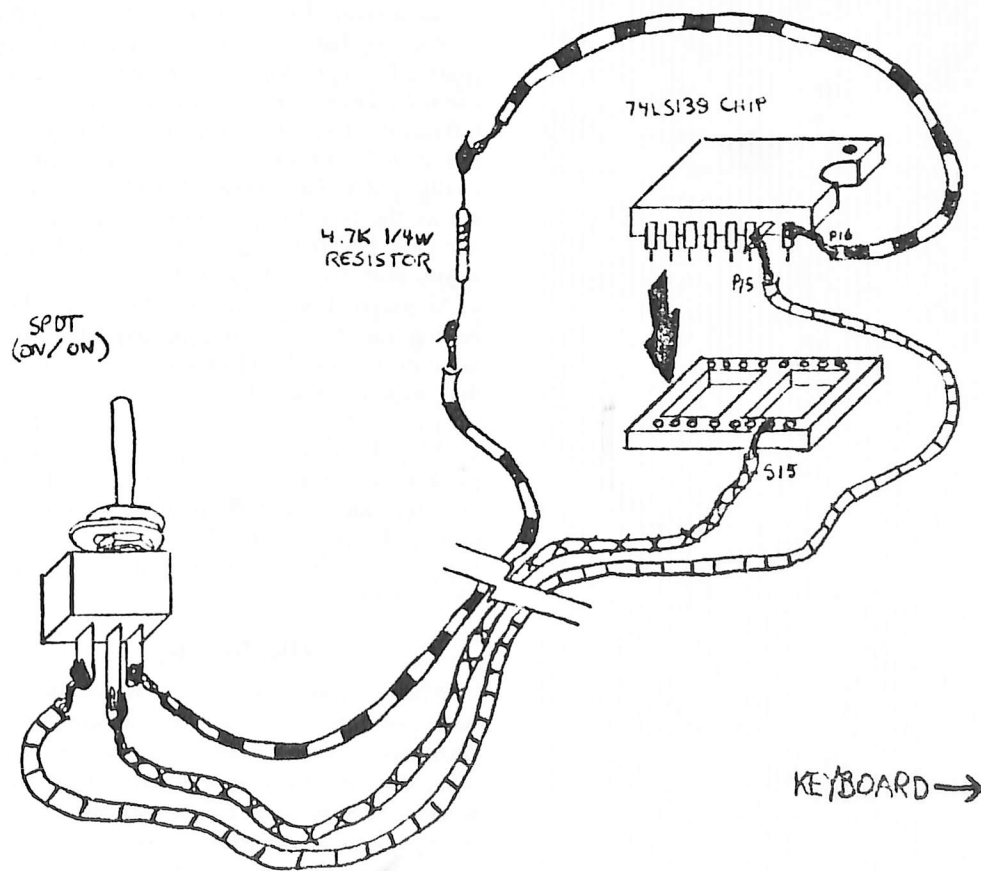
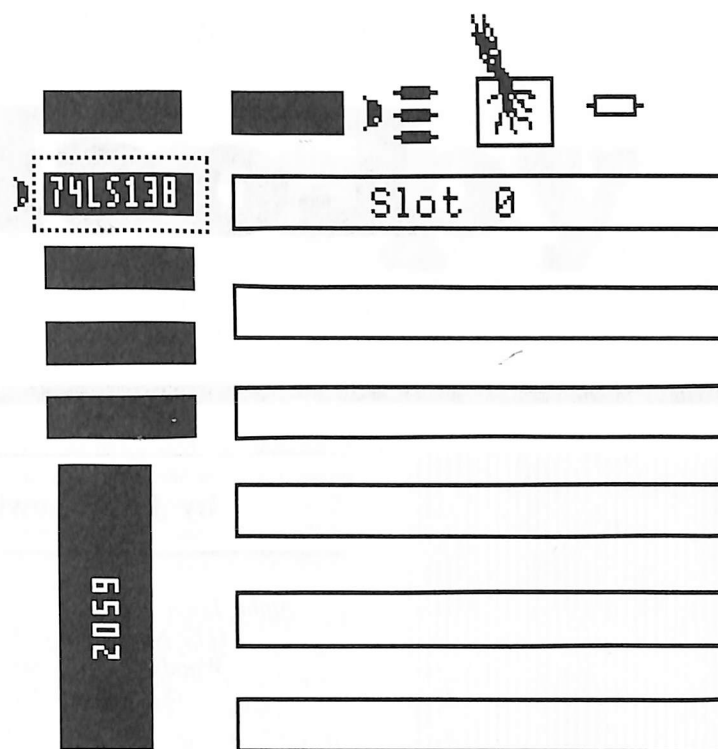
The above prompt change is not necessary to this example though.

C080 C080

This mode change is important if you are using your RAMcard for cracking purposes. It puts your RAMcard in a write protect mode so that the data in it, when written to, will not be changed. But, programs can still change the data. They do this by changing the RAMcard mode first from their own program. So, to keep programs from doing that you flip the switch you installed to the 'disable' position (don't forget to flip it back to 'enable' after you have used it for its particular purpose). Flip the switch now and try breaking out of some protected programs that work with 48K. It is the flipping of this SPDT switch that allows you to now have a "Protected Slot Zero."



Figure B



Locksmith

by Jerry Lewis

*Alpha Logic Business Systems, Inc.
4119 North Union Road
Woodstock, IL 60098
(815) 568-5166
\$79.95*

Requirements:

- Locksmith 6.0
- Blank initialized disk
- Method to create executable text file

The arrival of Locksmith 6.0 brings with it a new fast disk copy routine. The fast copy routine from previous versions can locate and use the memory expansion cards, but version 6.0 can also locate the large memory frequently found on the 80 column cards in the Apple //e auxiliary slot. The additional available memory allows the fastcopy routine to copy a disk in one pass. One pass copying is faster and easier if you only have one disk drive.

Although Locksmith 6.0 loads quickly, having the fastcopy routine available as a separate file would allow its presence on the disk with my other often used utilities. I read the procedure previously published in COMPUTIST No. 14 to capture the file from Locksmith 5.0, looking for some insight as to how the one from 6.0 could be captured. I quickly determined that, although the approach is good, version 6.0 does not run in the same memory location.

The Journey

When all else fails, there is always the fall back position, boot trace. As it turned out, getting the pieces for the fastcopy routine was relatively easy, the real problem was getting a working program. The following is the account of my journey to arrive at the step-by-step procedure at the end.

I first filled memory with \$22 at all locations from \$0800 to \$BFFF so that after the boot, I could tell what had come from the disk.

**CALL-151
FF59G N
0800:22
0801<0800.BFFEM**

I then moved boot zero from the disk drive card to RAM. This allowed me to alter the routine and control the boot of Locksmith 6.0 by causing it to exit to the monitor.

**8600<C600.C6FFM
86F7:59 FF**

I then booted Locksmith 6.0 with an 8600G, and when the monitor prompt appeared examined the memory \$0800 to \$08FF. The only thing thing that the routine starting at \$0801 did was to load the remaining data from track one. Four of the sectors are read directly in to the text page, \$0400 through \$07FF and accounts for why the menu selection appears so soon after booting. One sector from track one is read into page \$9E and the remainder of the track is read into memory from \$B600 through \$BFFF.

Looking through memory page \$08 for the exit instruction showed an indirect jump at \$083D, JMP (\$0026). I then added additional instructions to the boot code in memory to allow track one to load, but prevent the exit at \$083D. I controlled the exit by storing a RTS instruction at \$083D and doing a JSR rather than a JMP to \$0801 at the exit of the boot instructions. Upon return from \$083D, I jumped to the monitor:

**86F8:A0 60 8C 3D 08 20 01 08
8700:4C 59 FF**

After booting again and when track one had completed loading, I was back in the monitor. Examination of memory locations \$26 and \$27 showed that the exit at \$083D was to \$9E00. Looking through memory starting at \$9E00 showed that its primary purpose was to get the rest of the program into memory. There was an obvious exit at \$9E7A, JMP \$2000. Letting the rest of the program load and stopping the exit to \$2000 was the next order of business.

The instructions at page \$9E looked like they wouldn't overwrite my boot at page \$86. Therefore, another RTS at \$9E7A would work:

6.0 Fastcopy

8700:A9 60 8D 7A 9E 20 00 9E
8708:4C 59 FF

Once again, after booting Locksmith 6.0 (8600G), I was in the monitor. Starting at \$2000, I began to study what had just been loaded. Things were now getting more complex. However, since the program for selecting routines from the menu was now in memory, I now knew I was getting close.

The exit at \$9E00 to \$2000 carried the disk drive slot value in the X register, in the form of \$s0 (where s is the slot number). Thus by loading X with \$60 (my drives are in slot 6) and jumping to \$2000, Locksmith 6.0 should continue running. Since my boot program, starting at \$8600 was still intact, I just added on the necessary instructions:

870B:A2 60 4C 00 20

An 870BG started the drive and presented the menu selection screen. Selection of the Fast Backup option resulted in the program loading and operating correctly.

This ability to start the program at \$2000 gave me the opportunity to experiment and determine how the Fast Backup routine could be captured. Some place the program had to compare the key selected and take action based on the results. There was no obvious comparisons to "F" (\$C6), so I then looked for keyboard access.

I wanted to examine the memory starting at \$2000 before it was executed, so I had to go back and rerun the boot routine again. This is when I found that there is a routine, starting at \$2035, that places a zero at the first byte of each page, starting at page \$24 through \$B5. After rebuilding my boot routine at \$8600, and saving it this time, I booted Locksmith 6.0 again.

Stopping the exit to \$2000, left me in the monitor. Since I could find no references to \$C6 (F), I decided to look for references to the keyboard, \$C000. Using the search routine in the Apple //e enhanced monitor, I searched \$0800 to \$BFFF

C000<0800.BFFFS

Several references were located, but the one at \$1103 turned out to be the one used for menu selection. A search for references to \$1103 resulted in locating a routine, starting at \$128A, that went to get a key that was compared to values in a data table.

The data table is located in memory page \$0E and at \$0EB9 I found the value of \$C6, followed by four other bytes, prior to the byte for the next letter. The first two data bytes turned out to be what I had been looking for. They are pushed on the stack and an RTS resulted in executing a routine at \$1064 that loaded the Fast Backup routine from the disk.

When the disk access was completed, the routine was exited at \$1069 to the start of the Fast Backup routine at \$2002. I again loaded Locksmith 6.0, stopping the exit to \$2000. Before restarting the program at \$2000, with a \$870BG, I stored a jump to the monitor, JMP \$FF59, at \$1069

1069:4C 59 FF

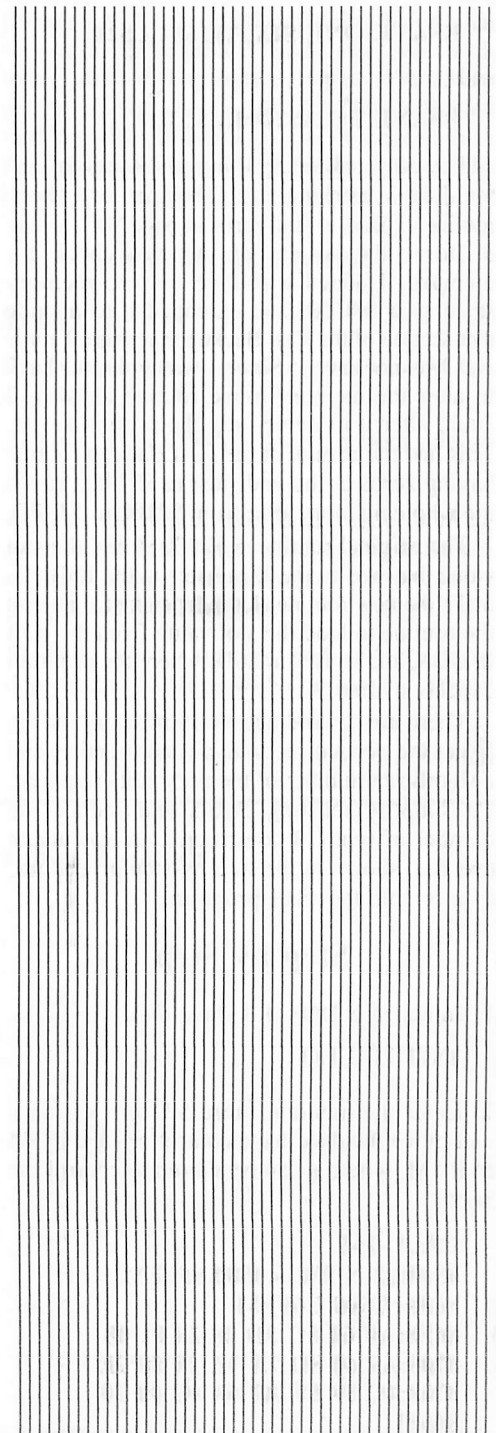
When Locksmith 6.0 was restarted and Fast Backup selected, the disk was accessed and I was greeted with a monitor prompt.


Just to make sure that what I had at \$2002, was what I wanted, I started the routine with a \$2002G. There, on the monitor, was the Fast Backup routine. Getting the routine back in memory again and examining it in detail, showed that getting a working copy was not going to be just a case of saving some memory.

One of the first things Locksmith 6.0 does, in the routine starting at \$2000, is check the slots for memory cards. The results of the slot examination is stored starting at \$0C93, and is moved to zero page by the Fast Backup routine. To prevent locking in the current memory card configuration, the routine to search the slots needed to be included in the final file.

The problem was that both the card check and Fast Backup routines were written to run in page \$20. There were also portions of the initial routine at \$2000 that were not needed. The only way I was going to get a compact routine, was to save all the pieces and then patch them together with the necessary address changes to enable them run at a new location.

... Continued on page 14



Now that I had some understanding of how Locksmith 6.0 worked and loaded the Fast Backup routine, I started all over from the beginning. Careful examination of the routine starting at \$2000, revealed that I only needed portions of the section from \$2000 through \$2330. I then booted a disk (6 ) and saved the file, BSAVE FC, A\$2000, L\$331. Starting again and this time jumping to the monitor at \$1069, I studied memory starting at \$2002.

There was a JSR at \$2008 to a small routine at \$1691 that needed to be saved. The actual Fast Backup routine is loaded in at \$2100 and the section from \$2100 through \$38FF is moved to \$0800 where it runs.

Examination of the routine showed that the memory beyond \$33DE is all zeros, thus does not really need to be saved. I then booted a disk

6 

and saved the two needed sections of memory:

BSAVE ONE SIX, A\$1684, L\$20

and

BSAVE FC, A\$2000, L\$13DF

The only remaining thing to do was to start building a working file. The memory check routine expects to find zeros in page \$0C, where it stores the results of the slot check for memory. I added a routine to zero memory starting at \$0C00, and extended it to include that section of the Fast Backup that was all zeros and was not saved. Since I was going to have to alter the memory move routine anyway, I could alter it to move only the portion saved. This had the results of the Fast Backup routine having zeros at the end, just as if I had saved the whole memory area from \$2100 to \$38FF.

The original routine at \$2000 stored the slot value that was in the X register in the IOB. To make the new routine slot independent, I altered the routine to load the slot value from the IOB in DOS, and store it at \$6D, where the program expects to find it.

There are several necessary memory moves to get the needed sections pieced together and a significant number of memory edits required. To simplify locating errors and reduce the number of typos, I constructed an executable text file, shown in the step-by-step procedure, to assemble the final program.

Step by Step

- 1) Initialize a DOS 3.3 slave disk.

INIT HELLO

- 2) Enter the monitor, store \$22 in memory and move the disk drive controller card ROM to RAM, where it can be altered to control the boot.

CALL-151

0800:22 N 0801<0800.95FEM

8600<C600.C6FFM

86F8:A0 60 8C 3D 08 20 01 08

8700:A9 60 8D 7A 9E 20 00 9E

8708:4C 59 FF A2 60 4C 00 20

86F8L

Check your typing against the following:

```
86F8- 10 60 LDY #560
86FA- 8C 3D 08 STY $083D
86FD- 20 01 08 JSR $0801
8700- A9 60 LDA #560
8702- 8D 7A 9E STA $9E7A
8705- 20 00 9E JSR $9E00
8708- 4C 59 FF JMP $FF59
870B- A2 60 LDX #560
870D- 4C 00 20 JMP $2000
```

- 3) Save the boot RAM to the slave disk in case of an error later (Yes, you can save from the monitor.)

BSAVE LS BOOT, A\$8600, L\$110

- 4) Place Locksmith 6.0 in your drive and boot with the code in RAM.

8600G

- 5) When the monitor prompt appears, boot the slave DOS 3.3 disk previously created. Do not use open-apple to boot, as it will cause memory to be altered.

6 

BSAVE LS START, A\$2000, L\$331

- 6) Place the Locksmith 6.0 disk back in the drive. Enter the monitor and boot Locksmith 6.0 again.

CALL-151

8600G

- 7) When the monitor prompt appears, edit memory to stop the execution of the Fast Backup routine.

1069:4C 59 FF

- 8) Restart the program and select the Fast Backup routine.

870BG

- 9) Select Fast Backup. When the monitor prompt appears, boot the DOS 3.3 slave disk and save the final two sections of memory.

6 

BSAVE FC, A\$2000, L\$13DF

BSAVE ONE SIX, A\$1684, L\$20

- 10) With your word processor or assembler, type in the following executable text file and save it on your slave disk as "FC BUILD."

```
MON C, I, O
CALL-151
0800:22 N 0801<0800.9000M
BLOAD LS START, A$4000
BLOAD ONE SIX, A$4484
BLOAD FC, A$5000
2014<400C.4011M
201A<402F.4034M
2025<4051.405DM
2035<4160.4330M
2206<5000.503FM
2246<503D.505AM
2264<4484.44A3M
2284<50FE.63DDM
2000:A0 00 98 99 00 0C C8 D0
2008:FA EE 05 20 AD 05 20 C9
2010:20 D0 ED 78
```

```
201B:35 20
201E:24 21 AD E9 B7 85 6D
2032:4C 06 22
2045:76 20
2086:09 21
20A9:0B 21
20CA:0B 21
20E7:09 21
20FA:4C 20
20FF:76 20
2104:76 20
2107:47 20
2141:CF 21
220D:71 22
2244:62
2246:3E A0 84 84 3C A0 00
224E:42
2254:22
2258:35
2273:64 22
BSAVE LS 6.0 FASTCOPY, A$2000, L$1563
3D0G
NOMON C, I, O
```

- 11) Start the text file executing.

EXEC FC BUILD

You should now have a working copy of the Fast Backup routine.

During the building of the separate Fast Backup file from Locksmith 6.0, I attempted to minimize the amount of memory saved. The reason for this is that I own a QuikLoader card from Southern California Research Group which uses EPROM to store files. Since I wanted to minimize the EPROM space used for the fast copy routine, I had to minimize the file size.

The card did present a problem though, Locksmith 6.0 will not run with the card installed. During the process of checking for memory cards, the quikLoader card is turned on, but doesn't know what to do, so it goes off into never-never land. The following patch will correct the problem for the quikLoader card, and may correct the problem with other peripheral cards.

Search the disk for the following byte sequence and replace it with the indicated change.

Currently:

```
0A 09 04 05 42 05 3D AA A5 42 9D 80
```

Change to:

```
18 65 42 05 3D AA 98 EA 09 10 9D 84
```

The addition of the following lines to the memory changes in the above executable text file will patch the fast copy file.

```
2110:18 65 42 05 3D AA 98 EA
```

```
2118:09 10 9D 84
```

Finally, if you wish to add Locksmith Fast copy to a hard disk (even though it won't copy hard disks), you will have to make it force the slot of your disk drives by adding the following line to the end of memory changes in the above executable text file:

```
2020:A9 60 EA
```



Graphics Expander

by Jim S. Hart

Springboard Software
7807 CreekrIDGE Circle
Minneapolis, MN 55435
\$39.95

Requirements:

Graphics Expander original
Super IOB or COPYA
Sector editor
Disk searcher
Two blank disks (or both sides of one disk)

The Graphics Expander, put out by the folks at Springboard, adds extra punch to your Print Shop by supplying 300 extra graphics. If you have used the Newsroom before, these graphics are a familiar sight (they are from the Clip Art disks). Also included is a graphics editor (again taken from the Newsroom) that allows you to edit graphics from either the Graphics expander disk or Print Shop graphics disks. Overall, the program is an excellent value but it has one drawback (if you don't know, then why are you reading COMPUTIST?): copy-protection.

Looking through my back issues of COMPUTIST, I came across Doni Grande's deprotection of the Newsroom (COMPUTIST No. 23) and thought to myself "Since both the Newsroom and the Graphics Expander are put out by the same company, why don't I use the Newsroom softkey to deprotect the Expander?". A good idea, but unfortunately it did not work all the way. Presented here are my additions to the Newsroom softkey so that it will work on the Graphics Expander. Those of you in a hurry can go to the Cookbook section while the rest of you can follow along...

What's Happening

The first thing I did was to copy the Expander with the Newsroom controller. Everything seemed to copy okay. However, when I booted the copy it didn't work. No amount of fiddling with that controller would get it to make a working copy. Deciding that there was some other protection hidden on the disk, I went about looking for direct disk access code (references to \$C088-\$C08F) that wasn't on the DOS tracks.

Using the sector editor in Copy II Plus, which has a search feature built in to it, offensive code was found in a couple of spots on the disk. Noting where it was on disk (on my disk the code was at track \$0E, sector \$0B and track \$18, sector \$02), I then did a disk map to find out which files contained the code. The files DSK and DGE contained code that looked for non standard DOS marks. I could now BLOAD the files and scan through them to find out the code's actual location in memory. It turned out that both routines started at location \$7B3A.

Now we are getting somewhere! All I had to do (or so I thought) was scan the disk for 20 3A 7B (JSR \$7B3A) and replace it with EA EA so that the routine was never called. I found one call in the file B.NEWS START (go ahead and look), which is the boot up program, and one in the file DGE. I now NOPed the calls and tried booting the disk. It still didn't boot correctly so that means I missed something. I put the JSRs back in and then looked at the code again. Notice what happens AFTER the JSR \$7B3A. Altering the code so that the disk check would seem successful and no bad jumps would be taken seemed to be the last barrier to overcome. I made the appropriate changes to the disk and then tried to boot up again. No luck. At this point I was stumped.

Reading over the Newsroom softkey once again, I noticed that if there was a nibble count on the disk I should search for the byte sequence 2C 50 C0 A9 01 85 FB 20 24 1F. Looking

through the disk once again, I found almost that exact string of bytes on track \$02, sector \$0F. The only difference is that there is a 27 in place of the 24. Making the one final (I hoped) change, as found in the "Change to:" list, I booted the disk up and everything worked like a champ! I later found out that the code in listings 1 & 2 was to first verify that the original was in the drive (JSR \$7B3A) and then check to see which disk (program, clip art, Print Shop graphics) was in the drive. The softkeyed version will still check to see which disk is in the drive and beep at you nastily if it isn't the one that is supposed to be in there.

Cookbook Method

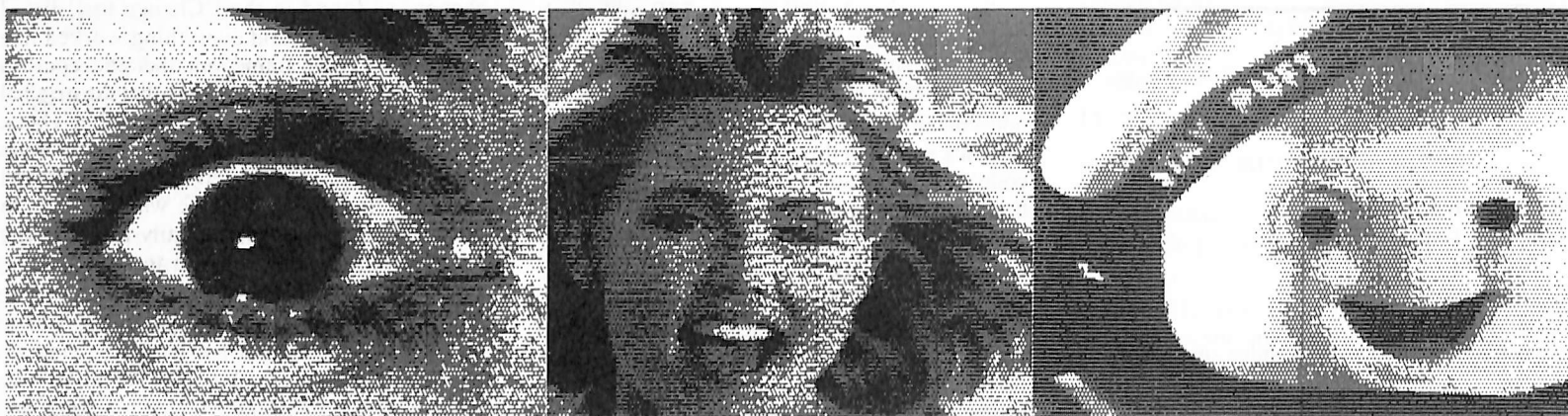
- 1) Boot up DOS 3.3.
- 2) Disable error checking with a POKE 47426,24.
- 3) Now, RUN COPYA and copy both sides to the blank(s).
- 4) After the copies have been made, boot up your sector editor/disk searcher and make the changes shown in the table below to the program side of the copy.
- 5) Hide your originals and use the now deprotected backups with confidence. Enjoy!



Sector Edits for Graphics Expander

Track	Sector	Byte(s)	From	To
\$00	\$02	\$9E	ED	DE
\$00	\$02	\$A3	BB	AA
\$00	\$03	\$35	ED	DE
\$00	\$03	\$3F	BB	AA
\$00	\$03	\$91	AD	DE
\$00	\$03	\$94-9B	DD 8C C0 D0 A9 4C F0 BC	EA BD 8C C0 10 FB C9 AA
\$00	\$06	\$F9	BB	AA
\$02	\$0F	\$09-0F	A9 01 85 FB 20 27 1F	A9 00 38 B0 09 EA EA
\$0B	\$06	\$37	23	2A
\$0B	\$06	\$46	14	1B
\$0B	\$06	\$4C	0E	15
\$18	\$02	\$B2-B5	D0 1E B0 1C	18 EA EA EA

Computer Eyes / 2



by **Gordie Lachance**

*Digital Vision Inc.
14 Oak Street - Suite 2
Needham, MA 02192
(617) 444-9040*

Okay, so you have a 1 megabyte RAM card, 40 megabyte hard disk, clock card with MIDI interface, 64K printer buffer, extended 80 column card, Accelerator card and a 1.21 gigawatt power supply. What could you possibly add to your hopelessly under-equipped system to make it better? The answer is simple, Digital Vision's new video digitizer: Computer Eyes/2.

Of all the hardware devices I have dealt with before, this is definitely one of the most exciting. With it, you can take any standard NTSC video signal and convert it into a picture displayable on the Apple.

Setup

Overall installation was a breeze. You don't have to cut any traces on the motherboard, jumper any chips or wait until a full moon. The device consists of a small card that can be plugged into any slot except zero. There are three cables coming out of the card with the markings V, C and M. A quick glance through the documentation was all it took to discover that the strange abbreviations on the cables stood for Video, Computer and Monitor. I therefore connected these cables to what they were named after and voila, the Computer Eyes/2 (CE/2) was installed.

Support Software

Digital Vision currently has two programs that support the CE/2. The software that comes with it is called "Computer Eyes System Software" (clever name). For an extra \$24.95, you can purchase a program called "Computer Eyes System Software - CE/2 Enhancement" (also a clever name). The latter does everything that the former does plus a whole lot more. I'll discuss that a little later.

The software is supplied on a disk that supports both DOS 3.3 and ProDOS (reminds me of the article "DOS to ProDOS and Back" in COMPUTIST No. 25) and is not copy protected. The program is very user friendly with all options appearing as "Magic Window" type menus. You don't even need to read the documentation to digitize pictures. However, reading the documentation will explain some of the unusual features of CE/2 and give you tips on making better pictures.





Digitizing Modes

The CE/2 system can take the same video signal and digitize it up to 5 different ways. They are:

1) Hi-res Single Level; This is the simplest of all modes. It digitizes the video on to a standard Apple hi-res screen (280 by 192) in monochrome. The single level means that a fixed threshold is used which results in a very high contrast picture (suitable for line art etc.).

2) Hi-res Multi Level; In this mode digitized images are stored on the normal Apple hi-res screen in monochrome. But, grey levels are synthesized by using a complex process called dithering. This mode doesn't work to well for line art or drawings, but if you don't have double hi-res, this is the best mode for digitizing "real-world" pictures.

3) Double Hi-res Single Level; This mode is exactly like "Hi-res Single Level" except images are stored on the double hi-res screen.

4) Double Hi-res Multi Level; This mode is exactly like "Hi-res Multi Level" except images are stored on the double hi-res screen.

5) Double Hi-res Gray Scale:

a) This mode produces by far the best images. However, in order to use it, you need a special 80 column card such as the Video-7 Enhancer series or the Legend Industries E'Card that can produce true gray scale. Because the graphics put out by these cards are superior to Apple's graphics, you naturally get a superior picture.

b) If you don't have a special 80 column card, this option will appear as "Double Hi-res False Color." In this mode, the CE/2 software produces a picture that is in true gray scale with a resolution of 140 by 192. To view the picture, you have to turn the color control of your monitor down so that the 16 double hi-res colors show up as 16 levels of grey instead. This mode is similar to the above mode, but it has much lower resolution.

File Formats

Standard hi-res pictures can be loaded in two ways either packed or unpacked. Although the packed method takes up less space and pictures tend to therefore load faster, the resulting files aren't compatible with any other program. The unpacked method is a straight BSAVE from the hi-res screen. You can't fit very many pictures on a disk this way but needless to say, this method is compatible with MANY programs.

For Double hi-res pictures, you have two more ways of saving the image. In addition to saving the file as packed or unpacked, you also have the option of saving it as as two files or one file. The two file method saves the main memory portion as a file and the auxillary memory portion as another file in a format that is compatible with Beagle Brothers Triple Dump and Beagle Graphics. The other format saves pictures in a format that is compatible with Dazzle Draw.

When loading a file, you do not have to specify what format it was saved in. The software is smart enough to figure that out by

itself. You just tell it the name, and the program will take it from there. I found this feature extremely time saving.

In addition to the above modes, if you buy the enhancement software, you get a bunch more file formats such as: Print Shop graphic and Newsroom photo. This then allows even more compatibility with other programs!

Digitizing Accuracy

Although best results were obtained when using a hi-res black and white camera on a tripod, I found that CE/2 could digitize a wide variety of video signals in an acceptable manner. Even color signals (like those from a VCR with a clear pause) were digitized quickly and effectively.

On the less enthusiastic side, I tried to digitize the output of another computer (an ACE 1000) and got less than favorable results. It seemed as if the video signal wasn't strong enough for the CE/2 to function properly. The CE/2 manual describes two trim pots on the card which may have helped it digitize the signal from the other computer but I didn't not wish to change these from their factory settings.

Summary

All things considered, I believe the Computer Eyes/2 is the best video digitizer you can get for the money. It should satisfy your black and white digitizing needs to their fullest extent.



"See ya."

"Not if I see you first!"



DOS to ProDOS and back

by Brian K. Chinn

Requirements:

DOS 3.3
ProDOS

In the time since "DOS to ProDOS and Back" was published (COMPUTIST No. 25, p.16-19), I have added some enhancements to my "DUAL DOS DISK MAKER" program. I have first replaced the cold, harsh "*** UNABLE TO LOAD PRODOS ***" message with a friendlier reminder that this disk is not bootable, for those who forget. The program now has very basic error checking capabilities, should you attempt to modify an unformatted disk, or something of that nature. Please keep in mind that the modifications that this program makes act upon existing ProDOS data on track 0, so the disk must be formatted for ProDOS use first, or it will not work.

For ease of use, I have tied the routine into the ampersand "&" vector so you can specify which drive, 1 or 2 (if you have a second drive), you want to use. Please note that the program gathers the drive parameter directly from the keyboard buffer, therefore it cannot be called from within a BASIC program via the "&" command.

The reason someone with two drives might want to use the DUAL DOS disk is if a program or file must be transferred often between the DOS and ProDOS environments, it is much easier to just leave the DUAL DOS disk in drive 2 and use it for storage under both operating systems. Such is the case with myself, since I prefer to write program code with a word-processor (primarily AppleWorks) save it as a ProDOS text file, convert it into a DOS 3.3 text file then assemble it with S-C Macro

Dual DOS Disk Maker 1.1
Source Code

```
*-----*
*          DUAL DOS DISK MAKER + PRODOS BOOSTRAP BUILDER *
*          UPDATE 1.1 *
*          BY: BRIAN K. CHINN *
*          DECEMBER, 1985 *
*-----*
          .OR $2000
          .TF DUAL DOS DISK MAKER
03D0- WARMDOS .EQ $03D0
03D9- CALLRWTS .EQ $03D9
03E3- SETRWTS .EQ $03E3
03F5- AMPERSAND .EQ $03F5
2300- BUFFER .EQ $2300
B7E8- DOSPARM .EQ $B7E8
C000- READKBD .EQ $C000
C010- SETKBD .EQ $C010
FDED- COUT .EQ $FDED
*-----*
*          SET UP '&' CALLING VECTOR *
*-----*
2000: A9 4C      PROG.START LDA #$4C
2002: 8D F5 03      STA AMPERSAND
2005: A9 10      LDA #BEGIN
2007: 8D F6 03      STA AMPERSAND+1
200A: A9 20      LDA /BEGIN
200C: 8D F7 03      STA AMPERSAND+2
200F: 60      RTS          ;DONE WITH SET UP
*-----*
*          FETCH '&' COMMAND AND READ DRIVE PARAMETER *
*          IF NOT A '2', THEN IGNORE IT - DRIVE = '1' *
*-----*
```

```

2010: A9 01   BEGIN   LDA #01   ;SET DISK DRIVE NUMBER 1
2012: 8D EA B7      STA DOSPARM+2
2015: AD 01 02      LDA $0201  ;READ THE NEXT CHARACTER
2018: C9 32          CMP #32    ;IS IT A '2'?
201A: D0 03          BNE BEGIN+15
201C: EE EA B7      INC DOSPARM+2 ;SET DISK DRIVE NUMBER 2
*-----*
*                MAKE THE DISK DOS COMPATIBLE                *
*                ADD CATALOG TRACK AND VTOC                   *
*-----*
201F: 20 DA 20      JSR CLEARBUF
2022: A2 80   DOSMAP LDX #80    ;T/S BITMAP BUFFER OFFSET
2024: A9 FF          LDA #FF    ;T/S BITMAP MARKING VALUE
2026: 9D 00 23      STA BUFFER,X ;ALLOCATE A TRACK AS FREE
2029: 9D 01 23      STA BUFFER+1,X
202C: E8            INX          ;SKIP THE NEXT FOUR BYTES
202D: E8            INX
202E: E8            INX
202F: E8            INX
2030: D0 F4          BNE DOSMAP+4
2032: A2 00   MAKEVTOC LDX #00    ;BUILD REMAINING VTOC AREA
2034: BD 60 21      LDA DATATBL1,X ;FETCH DATA FROM FIRST TABLE
2037: F0 0A          BEQ WRTVTOC ;TEST FOR 'EOD'
2039: BC 61 21      LDY DATATBL1+1,X ;FETCH DATA OFFSET
203C: 99 00 23      STA BUFFER,Y ;STORE DATA IN BUFFER
203F: E8            INX
2040: E8            INX
2041: D0 F1          BNE MAKEVTOC+2
2043: A9 00   WRTVTOC LDA #00
2045: 8D ED B7      STA DOSPARM+5 ;STORE SECTOR NUMBER TO WRITE
2048: A9 11          LDA #11    ;SET TRACK NUMBER TO WRITE
204A: 20 E6 20      JSR GORWTS ;WRITE VTOC
204D: 20 0C 21      JSR ERRCHECK ;CHECK FOR DISK ERROR
2050: 20 DA 20      JSR CLEARBUF
2053: A9 01          LDA #01    ;ZERO FIRST DIRECTORY SECTOR
2055: 8D ED B7      STA DOSPARM+5 ;STORE SECTOR NUMBER TO WRITE
2058: A9 11          LDA #11    ;SET TRACK NUMBER TO WRITE
205A: 20 E6 20      JSR GORWTS
205D: 20 0C 21      JSR ERRCHECK ;CHECK FOR DISK ERROR
2060: A9 11   BUILDDIR LDA #11
2062: 8D 01 23      STA BUFFER+1 ;STORE DIRECTORY TRACK
2065: A2 0F          LDX #0F
2067: 86 FF          STX $FF    ;PRESERVE SECTOR VALUE
2069: 8E ED B7      STX DOSPARM+5 ;STORE SECTOR NUMBER TO WRITE
206C: CA            DEX          ;POINT TO PREVIOUS SECTOR
206D: 8E 02 23      STX BUFFER+2 ;STORE SECTOR POINTER
2070: 86 FF          STX $FF    ;SAVE NEXT SECTOR VALUE
2072: A9 11          LDA #11    ;SET TRACK NUMBER TO WRITE
2074: 20 E6 20      JSR GORWTS ;WRITE DIRECTORY SECTOR
2077: 20 0C 21      JSR ERRCHECK ;CHECK FOR DISK ERROR
207A: A6 FF          LDX $FF    ;FETCH NEXT SECTOR VALUE
207C: E0 01          CPX #01    ;CHECK TO SEE IF WE'RE DONE
207E: D0 E9          BNE BUILDDIR+9
*-----*
*                TELL PRODOS TO ALLOCATE HALF THE DISK FOR DOS *
*                AND CHANGE THE LOADER BLOCK MORE FRIENDLY    *
*-----*

```

Assembler, or EXEC it into Applesoft. This way, I leave the disk in drive 2 alone and only swap the system disks in drive 1. A more plausible reason is you may want to create the disk on a system that has two drives, but will have to end up using the disk on a system with only one drive, somewhere else.

The Procedure

1) Enter the DUAL DOS DISK MAKER program, hexdump or source code, and save it on a DOS 3.3 disk.

2) Format a blank disk, or as many as you need, under the ProDOS operating system. You can use the FILER program on the ProDOS Users Disk, the Appleworks disk formatter, or any other suitable utility. You may name the disk anything you want, as long as it complies with the ProDOS rules.

3) Boot up any DOS 3.3 disk and install the program.

BRUN DUAL DOS DISK MAKER

4) Place the ProDOS disk(s), formatted from step 2, into the appropriate drive and call the program. Repeat as often as necessary. For simplicity, you will only be informed in the event of an error. If you receive the Applesoft prompt "]" after calling the routine, then everything has probably gone well.

& (or &2 for drive 2)

Note: The program only checks to see if the character entered after the "&" is a "2"; if not, it uses drive 1.

5) You now have a DUAL DOS disk which can be copied with COPYA or any other suitable copy program. Of course, making another one is just as simple.

6) When you wish to use the DOS User's Conversion Kit (DUCK), simply set the DOS 3.3 drive to the appropriate slot and drive, and set the ProDOS prefix. You're all set to convert.

One last note, since this modification allows you only half as much storage space as usual, I would recommend it only be used as a TRANSFER disk, unless your particular application does not require large amounts of disk space. If you're using an applications program or are programming, save the data, which will need to be converted, to both your master storage disk and to the DUAL DOS disk. If this is not possible, copy the appropriate files onto the DUAL DOS disk before you convert.

Dual DOS Disk Maker 1.1

```

2000: A9 4C 8D F5 03 A9 10 8D   $AC89
2008: F6 03 A9 20 8D F7 03 60   $AC54
2010: A9 01 8D EA B7 AD 01 02   $5D1E
2018: C9 32 D0 03 EE EA B7 20   $5015
2020: DA 20 A2 80 A9 FF 9D 00   $69EE
2028: 23 9D 01 23 E8 E8 E8 E8   $1413
2030: D0 F4 A2 00 BD 60 21 F0   $D167
2038: 0A BC 61 21 99 00 23 E8   $38F6
2040: E8 D0 F1 A9 00 8D ED B7   $45BD

```

... Continued on page 20


```

2048: A9 11 20 E6 20 20 0C 21 $6935
2050: 20 DA 20 A9 01 8D ED B7 $A2DB
2058: A9 11 20 E6 20 20 0C 21 $3E33
2060: A9 11 8D 01 23 A2 0F 86 $D0BD
2068: FF 8E ED B7 CA 8E 02 23 $D7DD
2070: 86 FF A9 11 20 E6 20 20 $0EB2
2078: 0C 21 A6 FF E0 01 D0 E9 $E5BB
2080: 20 DA 20 A2 0C 86 FF 8E $1E54
2088: ED B7 A9 00 20 E6 20 20 $377E
2090: 0C 21 A6 FF E8 E0 0F D0 $3566
2098: EC A9 FF A2 10 9D 00 23 $F31B

20A0: CA 10 FA A9 01 8D 00 23 $38BD
20A8: A9 03 8D ED B7 A9 00 20 $CA5B
20B0: E6 20 20 0C 21 20 DA 20 $14FA
20B8: A2 00 BD 76 21 D0 05 BC $F342
20C0: 77 21 F0 06 9D 00 23 E8 $266C
20C8: D0 F0 A9 00 8D ED B7 A9 $0713
20D0: 00 20 E6 20 20 0C 21 4C $EA3C
20D8: D0 03 A9 00 85 42 A9 23 $8ABA
20E0: 85 43 20 D6 B7 60 8D EC $0019
20E8: B7 A9 01 8D E8 B7 A9 02 $3C30

20F0: 8D F4 B7 A9 00 8D F0 B7 $F443
20F8: A9 23 8D F1 B7 A9 00 8D $B755
2100: EB B7 8D F3 B7 20 E3 03 $4B2E
2108: 4C D9 03 60 90 51 A2 00 $07EF
2110: AD F5 B7 F0 4A C9 10 D0 $950B
2118: 02 AD 18 BD 29 21 F0 06 $6628
2120: 20 ED FD E8 D0 F5 4C D0 $6025
2128: 03 8D 87 87 AD AD AD C4 $1992
2130: CF D3 A0 C9 AF CF A0 C5 $D962
2138: D2 D2 CF D2 AD AD AD 8D $654F

2140: 00 8D 87 87 AD AD AD D7 $CE80
2148: D2 C9 D4 C5 A0 D0 D2 CF $CFAD
2150: D4 C5 C3 D4 A0 C5 D2 D2 $F91B
2158: CF D2 AD AD AD 8D 00 60 $A045
2160: 01 31 01 37 03 03 0F 02 $9B60
2168: 10 35 11 01 11 30 23 34 $8EF1
2170: 7A 27 FE 06 00 00 01 A6 $9F93
2178: 2B BD 88 C0 8A 4A 4A 4A $ED30
2180: 4A 69 C0 85 FF A9 00 85 $F37D
2188: FE 8D 0C C0 8D 0E C0 20 $75BB

2190: 89 FE 20 93 FE 20 2F FB $177A
2198: 20 58 FC 20 84 FE A2 00 $0E6B
21A0: BD 40 08 F0 06 20 ED FD $0A21
21A8: E8 D0 F5 4E 10 C0 AD 00 $ECFE
21B0: C0 10 FB 6C FE 00 A0 A0 $ECDF
21B8: A0 D7 C5 CC C3 CF CD C5 $3882
21C0: A0 D4 CF A0 D4 C8 C5 A0 $B772
21C8: 04 15 01 0C 20 04 0F 13 $50BD
21D0: A0 C4 C1 D4 C1 A0 C4 C9 $1DED
21D8: D3 CB AE 8D 8D 8D A0 A0 $E89A

21E0: A0 A0 D4 C8 C9 D3 A0 C4 $1FE2
21E8: C9 D3 CB A0 C8 C1 D3 A0 $7326
21F0: CE CF A0 C4 CF D3 A0 CF $406C
21F8: C6 A0 C9 D4 D3 A0 CF D7 $BABA
2200: CE AE 8D 8D 8D A0 A0 A0 $1647
2208: D0 CC C5 C1 D3 C5 A0 C9 $3269
2210: CE D3 C5 D2 D4 A0 C1 A0 $A6BF
2218: C4 CF D3 A0 CF D2 A0 D0 $E5B5
2220: D2 CF C4 CF D3 A0 C4 C9 $12AA
2228: D3 CB 8D A0 A0 A0 A0 $5E25

2230: A0 A0 A0 A0 A0 A0 C1 CE $A2E3
2238: C4 A0 D0 D2 C5 D3 D3 A0 $BA97
2240: C1 CE D9 A0 CB C5 D9 AE $98B7
2248: 8D 00 00 00 $F69E

```

```

2080: 20 DA 20 PRODOS JSR CLEARBUF
2083: A2 0C LDX #0C ;CLEAR LOADER BLOCKS
2085: 86 FF STX $FF
2087: 8E ED B7 STX DOSPARM+5 ;STORE SECTOR NUMBER TO WRITE
208A: A9 00 LDA #00 ;SET TRACK NUMBER TO WRITE
208C: 20 E6 20 JSR GORWTS ;WRITE EMPTY BLOCKS TO LOADER
208F: 20 0C 21 JSR ERRCHECK ;CHECK FOR DISK ERROR
2092: A6 FF LDX $FF
2094: E8 INX
2095: E0 0F CPX #0F ;HAVE WE DONE THREE BLOCKS?
2097: D0 EC BNE PRODOS+5
2099: A9 FF PRODOSMAP LDA #FF ;ALLOCATE A TRACK AS FREE
209B: A2 10 LDX #10
209D: 9D 00 23 STA BUFFER,X ;STORE PRODOS BITMAP DATA
20A0: CA DEX
20A1: 10 FA BPL PRODOSMAP+4
20A3: A9 01 LDA #01
20A5: 8D 00 23 STA BUFFER ;STORE STATUS OF TRACK ONE
20A8: A9 03 LDA #03
20AA: 8D ED B7 STA DOSPARM+5 ;STORE SECTOR NUMBER TO WRITE
20AD: A9 00 LDA #00 ;SET TRACK NUMBER TO WRITE
20AF: 20 E6 20 JSR GORWTS ;WRITE PRODOS BITMAP
20B2: 20 0C 21 JSR ERRCHECK ;CHECK FOR DISK ERROR
20B5: 20 DA 20 LOADER JSR CLEARBUF ;BUILD PRODOS LOADER BLOCK
20B8: A2 00 LDX #00
20BA: BD 76 21 LDA BOOT.OBJ,X ;FETCH DATA FROM BOOT PROGRAM
20BD: D0 05 BNE LOADER+15 ;DID WE ENCOUNTER A ZERO?
20BF: BC 77 21 LDY BOOT.OBJ+1,X ;IF YES THEN LOOK NEXT BYTE
20C2: F0 06 BEQ WRTLOADR ;ANOTHER ZERO = 'EOD'
20C4: 9D 00 23 STA BUFFER,X ;STORE DATA IN BUFFER
20C7: E8 INX
20C8: D0 F0 BNE LOADER+5
20CA: A9 00 WRTLOADR LDA #00
20CC: 8D ED B7 STA DOSPARM+5 ;STORE SECTOR NUMBER TO WRITE
20CF: A9 00 LDA #00 ;SET TRACK NUMBER TO WRITE
20D1: 20 E6 20 JSR GORWTS ;WRITE LOADER BLOCK
20D4: 20 0C*21 JSR ERRCHECK ;CHECK FOR DISK ERROR
20D7: 4C D0 03 JMP WARMDOS ;RETURN CONTROL TO USER
*-----*
* MAIN SUBROUTINES *
*-----*
20DA: A9 00 CLEARBUF LDA #BUFFER
20DC: 85 42 STA $42
20DE: A9 23 LDA /BUFFER
20E0: 85 43 STA $43
20E2: 20 D6 B7 JSR $B7D6 ;CLEAR BUFFER FOR DOS WRITE
20E5: 60 RTS
20E6: 8D EC B7 GORWTS STA DOSPARM+4 ;STORE TRACK NUMBER TO WRITE
20E9: A9 01 LDA #01
20EB: 8D E8 B7 STA DOSPARM ;STORE RWTS TABLE TYPE
20EE: A9 02 LDA #02
20F0: 8D F4 B7 STA DOSPARM+12 ;STORE DISK FUNCTION
20F3: A9 00 LDA #BUFFER
20F5: 8D F0 B7 STA DOSPARM+8 ;STORE BUFFER LOCATION
20F8: A9 23 LDA /BUFFER
20FA: 8D F1 B7 STA DOSPARM+9 ;STORE BUFFER LOCATION
20FD: A9 00 LDA #00 ;PREPARE EXTRA DOS LOCATIONS
20FF: 8D EB B7 STA DOSPARM+3 ;STORE VOLUME MATCHING NUMBER
2102: 8D F3 B7 STA DOSPARM+11 ;STORE BYTE COUNT
2105: 20 E3 03 JSR SETRWTS ;SET UP RWTS PARAMETERS
2108: 4C D9 03 JMP CALLRWTS ;MAKE RWTS CALL

```

```

210B: 60          RTS          ;RETURN FROM RWTS CALL
210C: 90 51      ERRCHECK BCC ERR.END ;CHECK FOR ERROR
210E: A2 00          LDA #$00
2110: AD F5 B7      LDA DOSPARM+13
2113: F0 4A          BEQ ERR.END ;VERIFY ERROR
2115: C9 10          CMP #$10
2117: D0 02          BNE ERRCHECK+15 ;GET TYPE OF ERROR
2119: A2 18          LDX #$18
211B: BD 29 21      LDA ERR.DATA,X
211E: F0 06          BEQ ERR.EXIT
2120: 20 ED FD      JSR COUT          ;PRINT ERROR MESSAGE
2123: E8            INX
2124: D0 F5          BNE ERRCHECK+15
2126: 4C D0 03      ERR.EXIT JMP WARMDOS ;RETURN TO BASIC
2129: 8D 87 87      ERR.DATA .HS 8D8787
212C: AD AD AD
212F: C4 CF D3
2132: A0 C9 AF
2135: CF A0 C5
2138: D2 D2 CF
213B: D2 AD AD
213E: AD            .AS ----DOS I/O ERROR----"
213F: 8D 00 8D
2142: 87 87            .HS 8D008D8787
2144: AD AD AD
2147: D7 D2 C9
214A: D4 C5 A0
214D: D0 D2 CF
2150: D4 C5 C3
2153: D4 A0 C5
2156: D2 D2 CF
2159: D2 AD AD
215C: AD            .AS ----WRITE PROTECT ERROR----"
215D: 8D 00            .HS 8D00
215F: 60          ERR.END RTS
*-----*
*          DATA TABLE - VTOC INFORMATION          *
*-----*
2160: 01 31 01
2163: 37 03 03
2166: 0F 02      DATATBL1 .HS 0131013703030F02
2168: 10 35 11
216B: 01 11 30
216E: 23 34            .HS 1035110111302334
2170: 7A 27 FE
2173: 06 00 00            .HS 7A27FE060000
*-----*
*          PRODOS BOOTSRTP BUILDER          *
*          BY: BRIAN CHINN          *
*          JUNE, 1985          *
*          NOTE: OBJECT CODE RESIDES HERE AS A DATA TABLE          *
*-----*
2176: 01          BOOT.OBJ .HS 01
2177: A6 2B          LDX $2B
2179: BD 88 C0      LDA $C088,X ;TURN OFF THE DRIVE
217C: 8A            TXA
217D: 4A            LSR          ;CONVERT SLOT NUMBER
217E: 4A            LSR
217F: 4A            LSR
2180: 4A            LSR
2181: 69 C0          ADC #$C0 ;COMPUTE BOOT ADDRESS
2183: 85 FF          STA $FF ;SAVE IT IN ZERO PAGE
2185: A9 00          LDA #$00
2187: 85 FE          STA $FE
2189: 8D 0C C0      STA $C00C ;DE-ACTIVATE 80-COLS
218C: 8D 0E C0      STA $C00E ;NO ALT-CHAR SET
218F: 20 89 FE      JSR $FE89 ;MONITOR 'SETKBD'
2192: 20 93 FE      JSR $FE93 ;MONITOR 'SETVID'
2195: 20 2F FB      JSR $FB2F ;TEXT
2198: 20 58 FC      JSR $FC58 ;HOME
219B: 20 84 FE      JSR $FE84 ;NORMAL

```

```

219E: A2 00          LDX #$00
21A0: BD 40 08      GETCHAR LDA $0800+BOOT.MSG-BOOT.OBJ,X
21A3: F0 06          BEQ GETKEY ;TEST FOR 'EOD'
21A5: 20 ED FD      JSR COUT          ;OUTPUT CHARACTER
21A8: E8            INX
21A9: D0 F5          BNE GETCHAR
21AB: 4E 10 C0      GETKEY LSR SETKBD ;CLEAR KEYBOARD BUFFER
21AE: AD 00 C0      LDA READKBD ;WAIT FOR KEYPRESS
21B1: 10 FB          BPL GETKEY+3
21B3: 6C FE 00      JMP ($0FE) ;BOOT THE DISK
*-----*
*          DATA TABLE - BOOT UP MESSAGE          *
*-----*
21B6: A0 A0 A0
21B9: D7 C5 CC
21BC: C3 CF CD
21BF: C5 A0 D4
21C2: CF A0 D4
21C5: C8 C5 A0      BOOT.MSG .AS -" WELCOME TO THE "
21C8: 04 15 01
21CB: 0C 20 04
21CE: 0F 13            .HS 0415010C20040F13
21D0: A0 C4 C1
21D3: D4 C1 A0
21D6: C4 C9 D3
21D9: CB AE            .AS -" DATA DISK."
21DB: 8D 8D 8D            .HS 8D8D8D
21DE: A0 A0 A0
21E1: A0 D4 C8
21E4: C9 D3 A0
21E7: C4 C9 D3
21EA: CB A0 C8
21ED: C1 D3 A0
21F0: CE CF A0
21F3: C4 CF D3
21F6: A0 CF C6
21F9: A0 C9 D4
21FC: D3 A0 CF
21FF: D7 CE AE            .AS -" THIS DISK HAS NO DOS OF
            ITS OWN."
            .HS 8D8D8D
2202: 8D 8D 8D
2205: A0 A0 A0
2208: D0 CC C5
220B: C1 D3 C5
220E: A0 C9 CE
2211: D3 C5 D2
2214: D4 A0 C1
2217: A0 C4 CF
221A: D3 A0 CF
221D: D2 A0 D0
2220: D2 CF C4
2223: CF D3 A0
2226: C4 C9 D3
2229: CB            .AS -" PLEASE INSERT A DOS OR
            PRODOS DISK"
            .HS 8D
222A: 8D
222B: A0 A0 A0
222E: A0 A0 A0
2231: A0 A0 A0
2234: A0 A0 C1
2237: CE C4 A0
223A: D0 D2 C5
223D: D3 D3 A0
2240: C1 CE D9
2243: A0 CB C5
2246: D9 AE            .AS -" AND PRESS ANY KEY."
2248: 8D 00 00            .HS 8D000000
224B: 00
            PROG.END .END

```

Information Master

by Jim S. Hart

High Technology Software Products Inc.
P.O. Box 14665
Oklahoma City, OK 73113
\$150 (for DOS 3.3 version)

Requirements:

Information Master original (DOS 3.3 version)
Super IOB v1.5
Copy II Plus 5.x
(or FID and a sector editor)
Two blank disks

Information Master, along with its companion programs Data Master and Transit, come together to form a complete database system. Information Master is the core database program that allows you to enter, sort, print, and generally organize your data. Data Master will let you reorganize the formats created by Information Master into different ones if needed while Transit allows you to convert VisiCalc data files into ones compatible with the Information Master system. My friend, who owns this system, wanted to:

- 1) Be able to make backups easily so that the original could be hidden from damage,
- 2) Add a faster DOS so that the boot process and I/O time would be greatly reduced, and
- 3) Add a machine language sort subroutine so that the sorting process would not take so long (originally over an hour in some instances).

This sounds to me like a softkey was needed. I took up the challenge. After a long round of boot code tracing (which essentially led me nowhere) and checking the disk's format, I

finally came up with this softkey. The softkeyed disk has been tested pretty thoroughly and no bugs have come up. To be on the safe side, however, make a copy of your data disks before you try it. If any bugs pop up, send me a note via COMPUTIST and I'll see what I can do.

Step One

(Before you do anything else, initialize the two blank disks with

INIT HELLO

and then read on.)

Now, the first thing to do is listen to the booting of the disk. If you are new to this type of thing, practice listening to a normal disk's boot a couple of times to see what it sounds like. Information Master's boot sounds fairly normal. The next thing to do is use a program like Locksmith's "Quickscan" to determine what tracks may have valid data on them. Again, the Information Master (hereafter referred to as IM) disk seems to have valid data on tracks \$00 to \$22 (a normal whole disk).

The next thing to look at is a raw nibble dump of any track that may have data on it. I chose track \$12 as my track. Checking over the prologs and epilogs of both the Data and Address fields shows that all is normal except for the last byte of the data header. There is a \$DB there instead of the normal \$AD. Patching in this change to Copy II Plus' sector editor reveals that this is the only change. Strangely enough, if you now try to read in any sector from track \$11 (catalog track) an I/O error will result. Hmmm. Checking a nibble dump of Track \$11 shows it to be in normal DOS 3.3 format! I found out that only the files were protected. The track/sector lists, VTOC, and catalog were all in normal format. At this point, if you alter DOS to read in the protected

files, FID and Super IOB will bomb because of those normal sectors. How do we get around this?

Ignoring Errors

Throughout different issues of COMPUTIST are changes that can be made to DOS 3.3 to ignore or jump over different types of errors. I hate scrounging through all of my back issues so I decided to compile them into one chart for easy reference (Table 1). Any changes or additions you have are welcomed (remember to send them in to COMPUTIST!).

Step Two

Using Table 1, you can see placing the bytes \$29 00 starting at \$B8FB will tell DOS to ignore the last byte of the data header. This is the only change we will need (the rest of the bytes do not change between the protected and normal format).

Install the IM controller (Listing 1) into Super IOB now and copy the original to one of the blank disks. After it is finished, put away the original and continue on reading. Go into the monitor with a CALL -151 and make the change at \$B8FB. Type 3D0G to get back to BASIC and now catalog the IM disk (you could catalog the disk before but now you can supposedly load the files too). The first file is "INFORMATION MASTER" so let's try to load it in (insert the copied disk now):

LOAD INFORMATION MASTER

Ah ha! Another protection to conquer. No, don't get your copy of Integer BASIC out because using it will not work either. Scanning the catalog entries reveals the problem. Look at the byte right before the first character of the filename "INFORMATION MASTER." It should be a \$82 which denotes a locked

AppleSoft BASIC file (check out Worth and Lechner's *Beneath Apple DOS* for more on this) but instead it is a \$A2. No wonder we received such a nasty error message! If there is an \$A2 in front of the filename in any catalog sector (Track \$11, Sectors \$E-F) use your sector editor to change it to a \$82. If you find a \$22 instead, change it to a \$02. Leave the other two files (+ serial number file) alone for now. This changes the filetype so that normal DOS can load in the file. Now try to load in the "INFORMATION MASTER" file:

LOAD INFORMATION MASTER

It works this time! Before we go on any further, copy the files using FID or Copy II Plus from this disk to the lone remaining blank (but initialized) disk. After copying the files to the other disk, put the initial copy away because it isn't needed any more. Onwards!

Step Three

All that is needed now is to take out some CALLs from the BASIC files which access routines built into the original (and crazy) DOS. The only bad thing about this softkey is that it removes the "space left on disk" routine. I felt that this was an acceptable trade-off to have a normal disk and files. If you are really desperate for this function I suggest using a fast DOS that will print out how many free sectors there are during a CATALOG. I haven't had the chance to try Pronto DOS out yet but Diversi DOS works just fine. Anyways, after looking throughout the BASIC files I found the offensive code and took it out.

The one final task was to capture the picture that is shown when the original boots and add

a HELLO file. To do this, first boot your original. When the picture shows up, take out the original disk and replace it with the softkeyed disk (the one you just transferred all the files to) and press Control-Reset repeatedly until the disk boots or dumps you into the monitor (from which you can then boot anyways with a "C600G"). After the "FILE NOT FOUND" error pops up, save the picture with

BSAVE INFO.PIC, A\$4000,L\$1FF8

Now type in the short hello program from listing #2 and save it with

SAVE HELLO

That's it! You now have a softkeyed Information Master disk. Make any modifications you want and enjoy!

Step by Step

1) Boot up DOS 3.3 and initialize the two blank disks:

**INIT HELLO
DELETE HELLO**

(repeat for both sides)

2) Install the IM Controller (Listing 1) into Super IOB, RUN it, and copy the original IM disk onto the first blank (but initialized) disk. Do not format the backup.

When Super IOB is done copying the original, take it out of the drive and put it aside for now.

3) Boot up your sector editor and read in Track \$11, Sector \$0E of the copy.

4) Look at the byte that immediately precedes the first character of each of the filenames. If you find a \$A2, replace it with a \$82. If you find a \$22, replace it with a \$02. After making these changes write the sector back to disk and repeat this step for Track \$11, Sector \$0F of the copy.

5) Now use your file copy program and copy all of the files (except the serial # file) to the remaining blank disk. When you have finished, remove the disk that was copied onto by Super IOB and put it away. We do not need it anymore since the remaining disk is the almost-softkeyed version of the program.

6) We must now load certain files, take out the protection CALLs, and SAVE them back to disk.

7) Boot up the almost-softkeyed disk and after you get the error message, clear BASIC memory and then load up the CREATE / MODIFY program:

**NEW
LOAD CREATE/MODIFY**

8) Type in this line and then SAVE the file back to disk:

```
1024 HTAB (40 - LEN (A$) ) / 2 + .5 : PRINT A$
      : CALL - 868 : A$ = "CHECK^ CATALOG^ FOR^
      DISK^ SPACE" : HTAB (40 - LEN (A$) ) / 2
      + .5 : PRINT A$ : CALL - 868
```

**UNLOCK CREATE/MODIFY
SAVE CREATE/MODIFY**

9) Clear BASIC memory and load up the MENU file:

**NEW
LOAD MENU**

10) Type this line in and SAVE the file back to disk:

```
2123 HTAB (40 - LEN (A$) ) / 2 + .5 : PRINT A$
      : CALL - 868 : A$ = "CHECK^ CATALOG^ FOR^
      SPACE" : HTAB (40 - LEN (A$) ) / 2 + .5 :
      PRINT A$ : CALL - 868
```

SAVE MENU

11) Again, clear BASIC memory and then load up the SORT program:

**NEW
LOAD SORT**

12) Type in this line and SAVE the program back to disk:

```
260 QZ% = P%(1 .N0) : P%(1 .N0) = P%(L1 .N0)
      : P%(L1 .N0) = QZ% : I = I - M : IF I < 1
      THEN 250
```

**UNLOCK SORT
SAVE SORT**

13) All that's left to do is capture the hi-res picture and add a HELLO program. Type in the HELLO program from LISTING #2 and save it with:

SAVE HELLO

Table 1

How to tell DOS 3.3 to ignore certain bytes & errors.

Address Field

Ignore what	Address	From	To
Prolog (D5)	\$B954	C9 D5	29 00
(AA)	\$B95E	C9 AA	29 00
(96)	\$B969	C9 96	29 00
Epilog (DE)	\$B990	C9 DE	29 00
(AA)	\$B99A	C9 AA	29 00

Data Field

Ignore what	Address	From	To
Prolog (D5)	\$B8E6	49 D5	29 00
(AA)	\$B8F0	C9 AA	29 00
(AD)	\$B8FB	C9 AD	29 00
Epilog (DE)	\$B934	C9 DE	29 00
(AA)	\$B93E	C9 AA	29 00

Miscellaneous

Ignore what	Address	From	To
All epilog & checksum errors	\$B942	38	18
Data checksum	\$B92D	D0 13	18 60
Address epilog	\$B989	D0 B7	18 60
Ignore other errors	\$BE48	38	18

14) To capture the hi-res picture, take out your original and boot it up. When the picture is displayed, open the drive door, take out the original, and insert the softkeyed disk. Now, press Control-Reset until either the disk drive reboots or you are dumped into the monitor. If you get dumped into the monitor (indicated by the "*" prompt) type C600G to boot the computer. If, by some strange occurrence, you get into BASIC instead, type PR#6 to boot the computer.

15) You will now get a "FILE NOT FOUND" error (the computer is looking for the picture on disk and we haven't saved it yet). Let's save the picture:

BSAVE INFO.PIC, A\$4000, L\$1FF8

You are now all done. Adding a fast DOS will speed disk accessing up greatly. Diversi DOS seems to work just fine and I guess Pronto DOS would too. Enjoy the flexibility your softkeyed disk gives you.

controller

```
1000 REM INFO MASTER CONTROLLER
1010 TK = 3 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1015 POKE 47355 , 41 : POKE 47356 , 0
1020 GOSUB 490 : GOSUB 610
1030 GOSUB 490 : GOSUB 610 : IF PEEK (TRK ) =
      LT THEN 1050
1040 TK = PEEK (TRK ) : ST = PEEK (SCT ) : GOTO
      1020
1050 POKE 47355 , 201 : POKE 47356 , 173 : HOME
      : PRINT "COPY^ DONE." : END
```

controller checksums

1000 - \$356B	1030 - \$392C
1010 - \$2445	1040 - \$6DC5
1015 - \$278C	1050 - \$64C3
1020 - \$7077	

New Hello program

```
10 TEXT : HOME
20 PRINT : PRINT CHR$ ( 4 ) ; "BLOOD^
      INFO.PIC,A$4000"
30 POKE - 16302 , 0 : POKE - 16297 , 0 : POKE -
      16304 , 0 : POKE - 16299 , 0 : POKE - 16368
      , 0
40 FOR I = 1 TO 1000
50 IF PEEK ( - 16384 ) < 128 THEN NEXT : GOTO
      70
60 POKE - 16368 , 0 : I = 1001 : NEXT
70 PRINT : PRINT CHR$ ( 4 ) ; "RUN^
      INFORMATION^ MASTER"
```

checksums

10 - \$FB33	50 - \$2358
20 - \$5757	60 - \$8D81
30 - \$C4C6	70 - \$4B1B
40 - \$1CFD	

softkey for...

Certificate Maker

Jim S. Hart

*Springboard Software, Inc.
7808 Creekridge Circle
Minneapolis, MN 55435
(612) 944-3912
\$49.95*

Requirements:

Certificate Maker' originals
4 blank disks (or 2 double-sided)
Disk copier that can ignore read errors
DOS 3.3 sector editor

Certificate Maker, from the folks at Springboard, is a program designed to allow you to easily make certificates for all occasions. Whether it be outstanding achievement, simple recognition, or just for fun, certificates can be churned out by youngsters just as easily as adults.

This appears to be Springboard's first (I think) ProDOS release. This would seem to allow you to put it on your hard drive with ease but alas, it is protected. Even worse, in the introduction of the manual it says not to copy the master disk or certificate disk because it "COULD DAMAGE THEM"! It's time to get rid of THAT kind of protection once and for all.

The general ProDOS protection scheme is to have some type of signature checking (nibble counting, zero bit insertion) and to have the protection code in the first ".SYS" -type file. Not here. To be honest, I still don't know what the protection code is looking for but I have found a way to circumvent it. I could not seem to find any suspicious code on the disk using

my disk searcher so I simply went into the monitor (Control-Open Apple-Reset and then Control-Reset in rapid succession) and started snooping around.

At \$4200 I found some disk access code (LDA \$C0EC) mixed together with lots of comparisons so it was assumed this might be a protection scheme of some sort. Scanning the disk for 20 00 42 (JSR \$4200), I found it in only one spot: track 3, sector \$A, bytes \$78-7A. Disassembling the code around it revealed it indeed was the protection call. After making the changes below to circumvent the protection call, you will have a deprotected Certificate Maker to use so you can put those precious originals away. By the way, I broke into memory at various different times and I could not always find the code described above at \$4200. Self modifying, perhaps?

1) Using a copy program that can ignore errors on track \$22, copy the Program Master side onto one of the blank sides. To use Super IOB for this just change the variable LT (last track) in line 1010 from 35 to 34. After copying this side, use any normal whole disk copier and copy the other 3 sides onto the blanks.

2) Get out your favorite DOS 3.3 sector editor and make the changes listed below. Alternately, you could search the disk for the byte sequence 20 00 42 (it occurs only once) and replace it with 18 90 03.

Trk	Sector	Bytes	From	To
3	\$A	\$78-\$7A	20 00 42	18 90 03

That's it!

Elite

by Tim Scott

Firebird Software
Box No. 49
Ramsey, NJ 07446

Requirements:

- 48K slave disk
- A blank disk
- A way to reset into the monitor
- A good file copier (like Copy II Plus)
- A fast DOS (optional)

Elite is a futuristic space combat & trading game with superior three-dimensional graphics. You pilot a Cobra Mrk III spacecraft while avoiding space pirates and the GalCop's Vipers. The object of the game is to reach the title of "Elite" via trading and killing off the right people.

For most deprotectors, games provide the purchaser with two challenges: beating the game, and cracking the disk. I set out to do both and have accomplished one: cracking the disk.

The Protection

The protection is very, very strange. When I tried to copy the game with a quick copier, the first few tracks were unreadable, the second few were easily read, and I stopped the copier when it showed more unreadability with the third set. Being mostly a novice at cracking, I broke into the monitor (I use the Senior PROM) after the game loaded, moved the RWTS to \$1900, booted a 48K slave, and saved it off. I then flipped in the Super IOB new swap controller and got no good results.

After looking at the disk with Bag of Tricks' TRAX, I found that most of the first tracks were "damaged" and almost all of the ones after this didn't have any prologues or epilogues. So, I tried having COPYA ignore prologues and epilogues, which didn't work out too well either.

Then, after having spent a couple of hours working on it, I made a last ditch effort to have COPYA ignore everything. That, as expected, didn't work either.

After checking around, I found that the catalog was intact and could be viewed. Here are the files I found:

```
E[☐] LITE
SCRN
ELB1
E[☐] LA
S[☐] EC3
```

The [☐]s were unseen until I looked at the sector itself and saw them there.

To find out what was in E[☐] LITE (the only BASIC program on the disk), I booted the disk and broke into monitor after I saw the prompt. I moved code from \$800 to \$1900 and checked out the end of the program (held at locations \$AF & \$B0.) I then booted a 48K slave, replaced \$1900 to \$800 and reseted the end of the program. I could now list it (it was impossible to do it after breaking into the monitor and then going into BASIC, even though I restored \$D6 to \$00.) Here is what I found:

```
10 D$ = CHR$(4)
20 PRINT D$"BRUN E[☐] LA"
30 PRINT D$"BRUN S[☐] EC3"
```

Finding this out, I decided I could try to transfer all the files over by loading them into memory, booting a 48K slave, inserting a fast-dos'ed disk, and then saving them off, one by one. To begin with, I just typed in the above program and then INITED a disk with ProntoDOS in memory.

I was able to copy files SCRNM and ELB1 over from the original with Copy II Plus but with the others, I got sector errors. These were the ones I would have to do the hard way.

Now came the tough part, finding where the binary files E[☐] LA and S[☐] EC3 started and ended (thus finding the length of each.) I would boot the original, break into monitor, BLOAD the file at a specific address, check the first few bytes, BLOAD it again (having it BLOAD at its correct address), and then search for the bytes I had recorded down. When I found these, it meant that that was the beginning of the file. I had to do a lot of searching from there by mainly guess and checking bytes that didn't look right (a lot of \$FF's with a few \$6F's thrown in every so often.)

Doing this I found out that E[☐] LA started

at \$0A00 with a length of \$671E and that S[☐] EC3 started at \$2000 with a length of \$04FA. After this, it was all downhill. I BLOADed E[☐] LA (after breaking into the monitor on the boot), booted the 48K slave, inserted my INITed disk with E[☐] LITE, SCRNM, and ELB1 already on it, and BSAVED it off. I used the same procedure with S[☐] EC3.

The Procedure

1) Make a 48K slave disk by booting a DOS 3.3 disk and INITIALizing a blank one.

```
FP
INIT HELLO
DELETE HELLO
```

2) Make the copy disk by INITing it with the following program (a fast DOS is also preferable with this).

```
10 D$ = CHR$(4)
20 PRINT D$"BRUN E[☐] LA"
30 PRINT D$"BRUN S[☐] EC3"
```

```
INIT HELLO
```

3) Copy files SCRNM and ELB1 over to the copy with a file copy program (I use Copy II Plus).

4) Boot the original, and after the prompt is shown, break into the monitor, and BLOAD in E[☐] LA.

```
PR#6
```

(After you see the prompt, break into the monitor. Be quick!)

```
C0E8 (turns off drive)
A56EG (reconnects DOS)
BLOAD E[☐] LA
```

(Insert 48K slave.)

```
C600G
```

(Insert copy disk.)

```
BSAVE E[☐] LA, A$A00, L$671E
```

5) Use the same procedure as in 4) but use S[☐] EC3 as the file. It starts at \$2000, and has the length \$4FA.

6) The disk is now copyable.

Catalyst 2.0 &

by R. Wideman

Quark, Inc.
2525 West Evans, Suite 220
Denver, CO 80219
(303) 934-2211

Requirements:

- Disk copier that can skip tracks or ignore errors
- Nibble editor (optional)
- Disk editor that can write from memory buffer
- ProDOS
- DOS 3.3 slave disk with HELLO deleted
- A blank disk

Both Catalyst 2.0 and 3.0 use the same protection scheme, but it is modified slightly for each. To avoid confusion, Catalyst 2.0 will be used for a detailed examination of the scheme, but softkeys for both will be provided in a step by step fashion.

To begin an examination of any disk, one begins by booting it up. Doing so with Catalyst reveals that it is operating under Apple's ProDOS system. This is an encouraging sign since we know that the disk operating system (ProDOS) will not be modified, as is the case with Pascal. Chances are good that the disk can be copied without errors from the onset. But if a copy is attempted, it should be noted that read errors are encountered on tracks one and two, while the rest of the disk copies fine. This information tells us that these two tracks are possibly being used for some type of disk signature--a nibble count perhaps. To investigate our suspicions, a nibble dump of either track one or two should be able to reveal what type of signature or altered format is being used.

A nibble dump of track 2, should look like:

```
FE FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF
FF FF D5 AA 96 9A B3 B9
BB F9 AF 9E DE AA EB FF
FF FF FF FF FF D5 AA AD
F4 F2 9A 97 B7 B4 EF 96
```

This code shows that a normal address mark is still being used (D5 AA 96), as is a normal epilogue (DE AA) and data mark (D5 AA AD). But more importantly, it shows that the information for the volume, track, sector, and checksum have been radically altered! The new code is 9A B3 B9 BB F9 AF 9E. This code is different for a number of reasons. This area of data is normally encoded in a fashion called "4+4" (see *Beneath Apple DOS* or similar books for a description). This encryption technique limits the range of legal values which can be used. The only legal 4+4 values found in the string above are BB and AF. Also, to properly encode the volume, track, sector, and checksum numbers, eight values would be needed. There are only seven used here. It would not at all be unreasonable at this point to assume that Catalyst uses a special routine to read in tracks one and two which accommodates this modified format. So this is what one should start to look for.

Since Catalyst uses the ProDOS system, one can easily load up the same operating system and load in files off the Catalyst disk for examination. The purpose of this article is not to explain ProDOS, so a few steps will just be stated so as to further the discussion of the protection scheme. After booting ProDOS, load in Catalyst 2.0's start up file:

BLOAD CATALYST.SYSTEM,A\$2000,TSYS

This is the file that gets Catalyst going and is the most likely candidate for containing the desired code. Once this file is in memory, the rest of the search involves tracing the program itself. Without going into too much detail (and spoil all the fun?), the program basically relocates itself up into memory and loads more code in. There are only two routines of interest to be found, the first of which can be seen starting at address \$21E5.

**CALL-151
21E5L**

```
21E5- AD 3B B5 LDA $B53B
21E8- 49 6B EOR #$6B
21EA- 85 00 STA $00
21EC- A0 00 LDY #$00
```

```
21EE- A2 56 LDX #$56
21F0- CA DEX
21F1- 30 FB BMI $21EE
21F3- B9 00 B7 LDA $B700,Y
21F6- 5E 20 B6 LSR $B620,X
21F9- 2A ROL
21FA- 5E 20 B6 LSR $B620,X
21FD- 2A ROL
21FE- 45 00 EOR $00
2200- 91 0B STA ($0B),Y
2202- C8 INY
2203- D0 EB BNE $21F0
2205- 60 RTS
```

This routine converts raw data into normal data and stores it at the buffer pointed to by \$0B and \$0C. But it also has an instruction which does some decoding of the data before storing it (\$21FE). This is not found in a normal routine of this type, so a note of it is made to correct it in the softkey.

The routine that was initially sought is found at \$226F.

226FL

```
226F- AD 3A B5 LDA $B53A
2272- E9 01 SBC #$01
2274- 4A LSR
2275- D0 4C BNE $22C3
```

The location \$B53A contains the track number being read, so this routine will branch for every track except one and two. For these two tracks the routine continues with a special read routine that loads them into memory at \$2000-\$3FFF. It just so happens that this same routine shows up in part of the code that gets read in off the tracks, so it will need correcting in two spots. To do so requires that the data for tracks one and two be stored normally on the disk, and for this routine to be modified so that it always skips the special read routine and reads them in normally.

By now the plan of the softkey should be apparent. The protection scheme has been identified as has the way around it. What follows, therefore, is the step by step procedure.

Catalyst 3.0

Catalyst 2.0 Step by Step

1) Copy the disk and either ignore errors on tracks one and two, or skip these tracks entirely. Be sure, though, that the backup has these tracks formatted for future use.

2) Boot up ProDOS and load in the startup file:

```
BLOAD CATALYST.SYSTEM,A$2000,TSYS
```

3) Trace the program's operation up to the point where tracks one and two have been loaded in:

```
CALL-151
20FC:4C 59 FF
```

(was 4C FF B0, a jump to \$B0FF)

```
2000G
```

(The screen will fill with text title page and then stop again. At this point the program has relocated itself).

```
B15C:4C 59 FF
```

(was 6C 03 B0, an indirect jump to \$2000)

```
B0FFG
```

(data will be read in and the program will halt)

4) At this point the data for tracks one and two is stored in memory from \$2000 to \$3FFF. Track one is from \$2000 to \$2FFF and track two is from \$3000 to \$3FFF. You will probably want to save the data so you do not have to repeat steps 2 and 3 again. To do so, boot up your DOS 3.3 slave disk with HELLO deleted and save it out:

```
BSAVE MYFILE,A$2000,L$2000
```

5) Here is a point where you have an option. You need to somehow place the track data onto tracks one and two of the backup you made in step one. That is why a disk editor that can write from a memory buffer to disk is required. One such program is The Inspector, which is DOS 3.3 based. If you do use a DOS 3.3 based disk editor, note that the data from the tracks was read in with a different sector skewing table.

If you just write out the data to the tracks under DOS 3.3, they will be read back to memory in the wrong order by Catalyst. To correct this, change DOS 3.3's sector interleave table at \$BFB8 before writing out the track data:

```
BFB8:00 02 04 06 08 0A 0C 0E
:01 03 05 07 09 0B 0D 0F
```

This is the table used to read the data in, so now it will be written out in the correct order.

Another option is to use the all-purpose Super IOB to write the sectors out for you. Install the controller following this article into Super IOB 1.5, have the binary file of the saved sectors on your Super IOB disk, and let 'er rip.

6) With any normal DOS 3.3 disk editor (with a normal DOS sector interleave table in memory), make the following sector edits. The first two remove the check for tracks 1 and 2, and the last deletes the data decoding (the EOR \$0D at \$21FE).

Track	Sector	Byte	Change to
\$02	\$0C	\$F7	A9 01 EA
\$0A	\$01	\$72	A9 01 EA
\$0A	\$04	\$FE	EA EA

Catalyst 3.0 Step By Step

1) Copy the Catalyst 3.0 disk, same as for the 2.0 disk, skipping or ignoring tracks 1 and 2.

2) Boot up ProDOS and load in the startup file:

```
BLOAD CATSTART.SYSTEM,A$2000,TSYS
```

3) Trace the program's operation up to the point where tracks one and two have been loaded in:

```
CALL-151
2175:4C 59 FF
```

(was 4C 00 40, a jump to \$4000)

```
2175G
```

(data will be read in and program will halt)

4) At this point the data for tracks one and two is stored in memory from \$4000 to \$5FFF. Track one is from \$4000 to \$4FFF and track

two is from \$5000 to \$5FFF. You will probably want to save it so you do not have to repeat steps 2 and 3 again. To do so, boot up your special DOS 3.3 disk with HELLO deleted and save it out:

```
BSAVE MYFILE,A$4000,L$2000
```

5) Write the saved sectors to tracks 1 and 2 using a sector editor or Super IOB in the same manner as for Catalyst 2.0.

6) With any normal DOS 3.3 disk editor (with a normal DOS sector interleave table in memory), make the following sector edits. The first two remove the check for tracks 1 and 2, the last two remove the decoding instruction.

Track	Sector	Byte	Change to
\$01	\$0C	\$2C	A9 01 EA
\$1F	\$06	\$06	A9 01 EA
\$1F	\$07	\$09	EA EA EA
\$1F	\$07	\$1E	EA EA

This concludes the softkey for Catalyst!

controller

```
1000 REM CATALYST
1010 TK = 1 : LT = 3 : ST = 15 : LS = 15 : CD = RD
: FAST = 1
1020 POKE 49080 , 15 : POKE 49095 , 0
1030 GOSUB 490 : GOSUB 610
1040 POKE 49080 , 0 : POKE 49095 , 15
1050 HOME : PRINT "TRACK^ 1^ &^ 2^ DATA^
WRITTEN" : END
10010 PRINT CHR$ (4) "BLOAD^ MYFILE,A$2700"
```

controller checksums

1000 - \$356B	1040 - \$ADD5
1010 - \$90C3	1050 - \$FB16
1020 - \$A51F	10010 - \$F8D3
1030 - \$7AF5	

Sword of Kadash and Rescue Raiders

by Russell Meyer

Sword of Kadash available from:
Penguin Software
830 Fourth Ave.
P.O. Box 311
Geneva, IL 60134
\$29.95

Rescue Raiders available from:
Sir-Tech Software
6 Main Street
Ogdensburg, NY 13369
\$34.95

Requirements:

At least 64K RAM
DOS 3.3
Rescue Raiders (unprotected)
Sword of Kadash (unprotected)

Note: Sofkeys for Rescue Raiders and Sword of Kadash appeared in COMPUTIST No. 16 and 27 respectively.

Both Rescue Raiders And Sword Of Kadash are great games. Rescue Raiders is the MOST addicting game I have ever played, but both have their drawbacks.

Rescue Raiders

Rescue Raiders's drawback seems to be the lack of bombs and guided missiles. With the tips by Clay Harrell in COMPUTIST issue 16, by typing "POPPY" or whatever is written backwards on track \$F sector \$9 bytes \$A8 to \$AC, 6 new commands would become available for use:

/: Free copters up to a maximum of 255
J: Moves you to the farthest left of the playing field
K: Moves you to the middle of the playing field
L: Moves you to the farthest right of the playing field
Return: Makes you bullet proof
-1: Pressing the "-" key followed by a number 1 to 9 will end your current battle and put you at that desired battle (level). Number 9 ends the game.

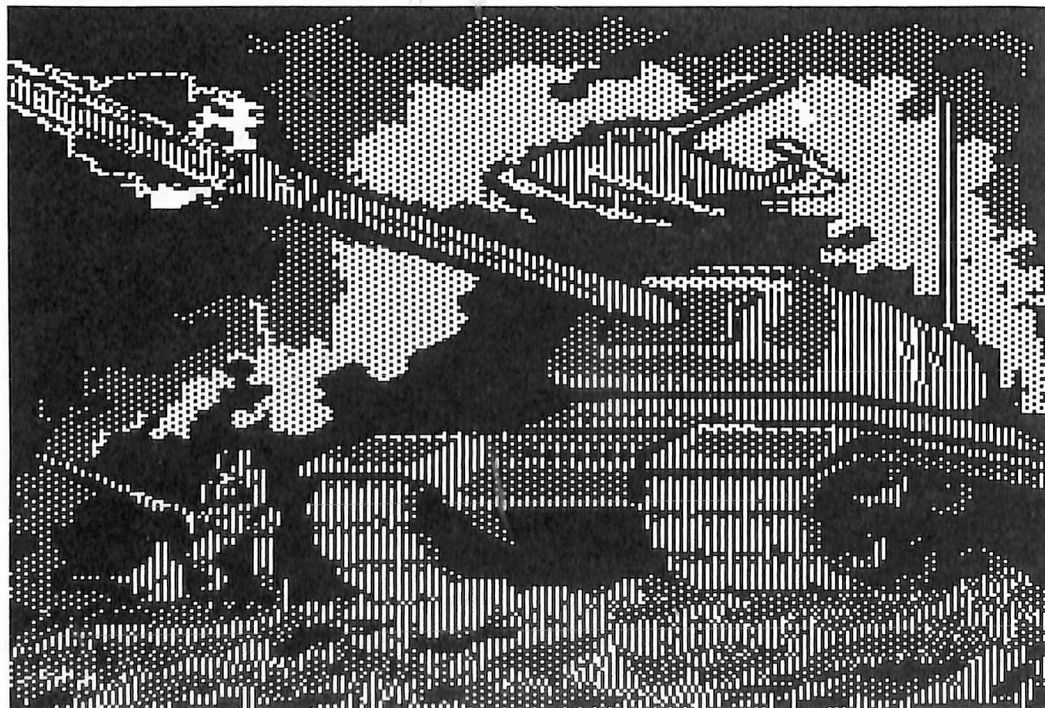
These commands were great but I found myself refueling and rearming my helicopter most of the time. So I decided to do a little disk snooping to find out where Rescue Raiders stored the data on the amount of arms the copter gets when it first starts out. I found what I

wanted on track \$D sector \$0.

Bytes \$50,\$5F,\$64 hold the number of bombs, rockets of napalm (Available when you get to the 5th level) and guided missiles respectively, in hex form. You can make these changes with a sector editor, but I have a few friends who don't know how to use a sector editor and were afraid to try. So I came up with a little program that anyone can use.

Rescue Raiders patcher

```
10 REM BY RUSSELL MEYER
20 HOME :RW = 1 :LO = 24576 :TF = 256
30 PRINT : PRINT "^ RESCUE^ RAIDERS^ A.P.T."
40 PRINT "^ PLEASE^ ENTER^ RAIDERS^ DISK"
50 INPUT "^ AND^ HIT^ RETURN." ;AS : HOME
```



```

60 POKE 768 ,32 : POKE 769 ,227 : POKE 770 ,3 :
   POKE 771 ,76 : POKE 772 ,217 : POKE 773
   ,3
70 POKE 47083 ,0 : POKE 47084 ,13 : POKE 47085
   ,0 : POKE 47088 ,LO - INT (LO / TF ) * TF
   : POKE 47089 , INT (LO / TF ) : POKE 47090
   ,0 : POKE 47091 ,0 : POKE 47092 ,RW
80 CALL 768 :B = PEEK (24656 ) :N = PEEK
   (24671 ) :GM = PEEK (24676 )
90 IF RW = 2 THEN END
100 PRINT " ^ YOU ^ HAVE ^ " ;B ; " ^ BOMBS ."
110 INPUT "HOW ^ MANY ^ WOULD ^ YOU ^ LIKE ^
   (0-255) ^ " ;B
120 PRINT : PRINT " ^ YOU ^ HAVE ^ " ;N ; " ^
   ROCKETS ^ OF ^ NAPALM ."
130 INPUT "HOW ^ MANY ^ WOULD ^ YOU ^ LIKE ^
   (0-255) ^ " ;N
140 PRINT : PRINT " ^ YOU ^ HAVE ^ " ;GM ; " ^
   GUIDED ^ MISSILES ."
150 INPUT "HOW ^ MANY ^ WOULD ^ YOU ^ LIKE ^
   (0-255) ^ " ;GM
160 VTAB 15 : PRINT " ^ HIT ^ RETURN ^ TO ^ SAVE ^
   CHANGES ^ " ; : GET AS$
170 POKE 24656 ,B : POKE 24671 ,N : POKE 24676
   ,GM :RW = 2 : GOTO 60

```

checksums

10	- \$BADD	100	- \$41D3
20	- \$B976	110	- \$3684
30	- \$A72F	120	- \$E8CB
40	- \$C9B1	130	- \$225D
50	- \$CFB9	140	- \$9860
60	- \$4339	150	- \$98D6
70	- \$6DC9	160	- \$57E1
80	- \$1C0D	170	- \$A1AB
90	- \$BE81		

Line by line summary

10-50: setup
60-70: loads track 13, sector 0 and stores the sector beginning at 24576
80-150: PEEKs needed values and asks the user for new values
160-170: POKEs the new values back and saves the sector

Sword of Kadash

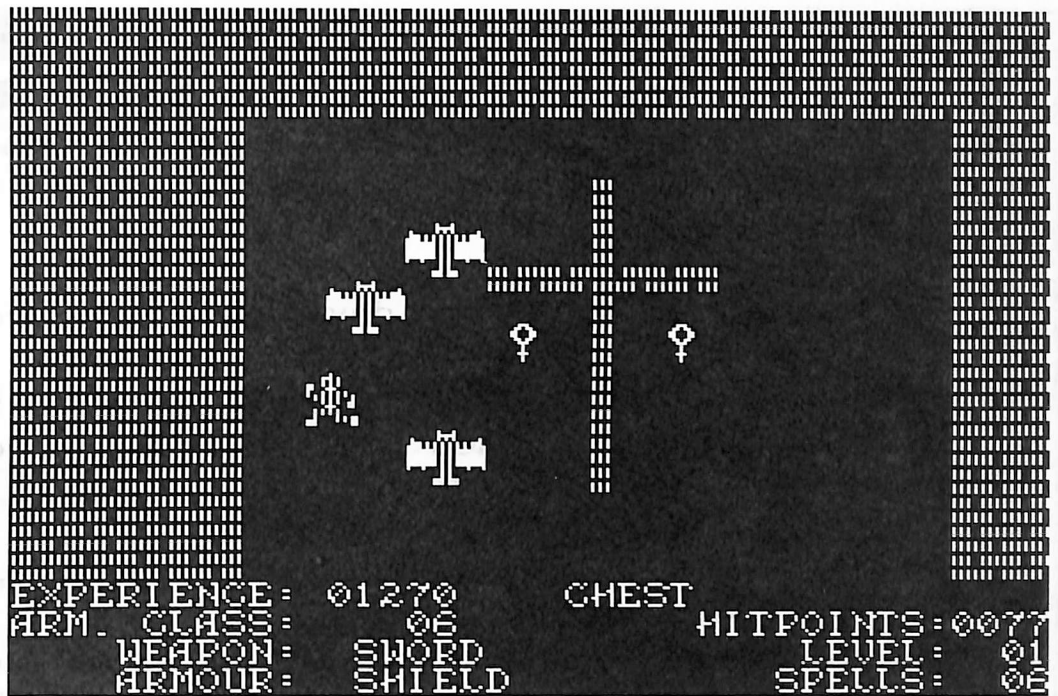
Sword Of Kadash is the other game that has a small problem. To play the game you need to make a character disk. Then if you get killed three times (it is easy to get killed three times) you have to make another character disk. It would be much more fun if you could get a few more men in the game (and stay alive a little longer), so after a bit of snooping on the character disk I found where the game stores the saved game data. The number of lives the saved character has left is stored on track \$22 sector \$F byte \$1A. Again you could change the byte with a sector editor but again my friends wanted an easier way.

Sword of Kadash program

```

10 REM BY RUSSELL MEYER
20 HOME :RW = 1 :LO = 24576 :TF = 256
30 PRINT : PRINT " ^ SWORD ^ OF ^ KADASH ^
   A . P . T ."

```



```

40 PRINT " ^ PLEASE ^ INSERT ^ KADASH ^
   CHARACTER ^ DISK ^ "
50 INPUT " ^ AND ^ HIT ^ RET. ^ " ;AS$ :HOME
60 POKE 768 ,32 : POKE 769 ,227 : POKE 770 ,3 :
   POKE 771 ,76 : POKE 772 ,217 : POKE 773
   ,3
70 POKE 47083 ,0 : POKE 47084 ,34 : POKE 47085
   ,15 : POKE 47088 ,LO - INT (LO / TF ) *
   TF : POKE 47089 , INT (LO / TF ) : POKE
   47090 ,0 : POKE 47091 ,0 : POKE 47092 ,RW
80 CALL 768 :MEN = PEEK (24602 )
90 IF RW = 2 THEN END
100 PRINT " ^ NUMBER ^ OF ^ MEN ^ YOU ^ HAVE ^ NOW ^
   IS ^ " ;MEN
110 INPUT "PLEASE ^ ENTER ^ THE ^ NUMBER ^ OF ^
   MEN ^ YOU ^ WOULD ^ LIKE ^ (1-255) ^ " ;MEN
120 POKE 24602 ,MEN :RW = 2
130 PRINT : PRINT " ^ HIT ^ RETURN ^ TO ^ SAVE ^
   CHANGES ^ " ; : GET AS$ : GOTO 60

```

checksums

10	- \$BADD	80	- \$2B28
20	- \$B976	90	- \$0586
30	- \$CCF5	100	- \$754D
40	- \$CFE2	110	- \$BA6D
50	- \$15A9	120	- \$DB3E
60	- \$E7E3	130	- \$0F9D
70	- \$DE69		

Line by line summary

10-50: set up
60-70: loads track 34, sector 15 and stores the seceginning at 24576
80-110: PEEKs the needed value and asks the user for new value
120-130: POKEs the new value back and saves the sector

This program only works on a character disk with a saved game on it. One thing you can do is make lots of copies of your saved character disk so you can restart from a certain place in the game over and over again.

Ultimaker IV

tip by Cracker Jack Mac

Requirements:

Apple II Plus or better
 Ultima IV
 A sector editor

In his "Ultimaker IV" article (COMPUTIST No. 30, p.16), Danny Pollak failed to mention how to edit each of your eight virtues. Simply use a sector editor to read track \$14, sector 3. The first four bytes are the last part of character eight from track \$14, sector 4. The next eight bytes, bytes \$4-\$B, control how well you are doing in each of the eight virtues. The following table lists the byte number and the virtue each one controls.

Byte	Virtue
\$04	Honesty
05	Compassion
06	Valor
07	Justice
08	Sacrifice
09	Honor
0A	Spirituality
0B	Humility

The value of these bytes range from 00-99 (Binary Coded Decimal). 01 is the worst and 00 is the best. To receive partial avatarhood in a virtue simply replace the value of the appropriate byte with 00 and write the sector back to the disk.

By the way, if you think you're really good at Karateka, boot the back side of your original Karateka disk.

The Book Of Softkeys

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

From deep within the COMPUTIST archives comes a collection of softkeys originally contained in issues 1 through 15. These volumes also contain some of the more useful programs and tutorials presented in those early issues. The books make an economical alternative to those rare (and unavailable) back issues of Hardcore COMPUTIST.

**SAVE
\$5**

Volume I: Issues 1-5

~~(\$12.95)~~ (\$7.95)

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

**SAVE
\$5**

Volume II: Issues 6-10

~~(\$17.95)~~ (12.95)

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

Volume III: Issues 11-15

(\$17.95)

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

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

  _____ Exp. _____

Signature _____ CP43

- Volume I \$7.95 + \$2 shipping & handling
- Volume II \$12.95 + \$2 shipping & handling
- Volume III + 17.95 + \$2 shipping & handling
- All three volumes! \$38.85 + \$4 shipping & handling

Foreign orders (except Canada & Mexico), please add \$5.00 per book shipping & handling. U.S. funds drawn on U.S. banks only. Most orders shipped within 5 working days, however please allow 4-6 weeks delivery for some orders. Washington State orders add 7.8% sales tax.

Send your orders to:

COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

Universal

AAA International

5 1/4 inch Wooden Diskcase
(Holds 70 5 1/4 inch disks).....\$20.25
3 1/2 inch Wooden Diskcase
(Holds 30 3 1/2 inch disks).....\$18.00

Abati

Power Strip (6 ft. cord).....\$16.50
Letter-Quality Parallel Printer (LQ-20P)
132 COL, 18 CPS, with tractor....\$298.50

Acrylic Arts, Inc.

80 COL Acrylic printer stand.....\$38.50
Universal printer stand.....\$17.00

Amdtek

Amdisk I - 3" disk drive.....\$274.50
Color II/S 13" hi-res monitor.....\$375.00
Video 300A Amber monitor.....\$183.00
Video 300G Green monitor.....\$172.50
Tilt/Swivel for color 300-730.....\$30.00
Letter-Quality Parallel/Serial printer
132 COL, 40 CPS.....\$945.00
DXY-100 XY plotter.....\$712.50

Anadex

Dot Matrix - Parallel Printer (DP-9620B)
132 COL, 240 CPS, graphics...\$1464.75

AST Research-Apple Division

Multi I/O (with
clock, calendar, 2 serial ports)....\$157.50
Multi I/O (with clock & calendar)..\$130.50

C H Products

Mach III w/Fire Button (I/e).....\$44.00
Mach II Joystick (I/e).....\$36.50
Mach III w/Fire Button (I, I+, I+).....\$44.00
Mach II Joystick (I, I+, I+).....\$36.50

Compu Cable

Apple II 6 ft parallel cable.....\$30.00

Hayes Microcomputer Products

Smartmodem 300
w/Smartcom I (I/c).....\$222.00
Micromodem I/e w/Smartcom I....\$186.00

Kraft Systems

Apple Joystick (3 buttons).....\$42.75

Micro Sci

80/64E 80 Column 64K Card (I/e).....\$51.00

Panasonic

Dot Matrix parallel printer (KX-P1092)
80 COL, 180 CPS.....\$454.50
Dot Matrix parallel printer (KX-P1081)
80 COL, 120 CPS.....\$298.50
Dot Matrix parallel printer (KX-P1091)
80 COL, 160 CPS.....\$389.25
Dot Matrix parallel printer (KX-P1592)
132 COL, 180/38 CPS.....\$627.00
Dot Matrix parallel/serial printer (KX-P1595)
132 COL, 240/51 CPS, 15K buffer\$816.00

Video-7

Color Enhancer I/e w/Dazzle Draw...\$162.00
Color Enhancer I/c w/Dazzle Draw..\$117.00

Books

Ashton-Tate

Intro to DBASE II.....\$20.00
Using DBASE II.....\$26.25

Educational Systems

Microref for Wordstar.....\$12.00
Microref for Supercalc.....\$12.00
Microref for Visicalc.....\$12.00

Microsoft

The Printed Word.....\$16.50
Appleworks.....\$15.75

Quality Software

Beneath Apple ProDos.....\$16.00

Software

Academy Software, Inc

Typing Tutor + Word Invaders (I/e)....\$16.00

How To Order

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

SoftKey Publishing PO Box 110816-T Tacoma, WA 98411

Name _____ ID# _____
Address _____
City _____ State _____ Zip _____
Country _____ Phone _____
Exp. _____
Signature _____ CP43

Software cont.

Accolade

Sundog.....\$28.50
Dambusters.....\$24.50
Hardball.....\$24.50
Fight Night.....\$24.50
Law of the West.....\$24.50

Activision

Greeting Card Maker.....\$32.00
Term Paper Writer.....\$48.00
Writer's Choice.....\$39.75
Filer's Choice.....\$39.75
Planner's Choice.....\$39.75
Hacker.....\$32.00
Great American Road Race.....\$32.00
Ghostbusters.....\$32.00
Tass Times in Tonetown.....\$32.00
Space Shuttle: Journey into Space...\$32.00
The Rocky Horror Show.....\$27.75
Shanghai.....\$32.00
Little People Discovery Kit.....\$32.00
Countdown to Shutdown.....\$32.00
Gamemaker.....\$39.75
Labyrinth.....\$32.00
Borrowed Time.....\$32.00
Portal.....\$36.00
Alter Ego (male).....\$39.75
Alter Ego (female).....\$39.75
Star League Baseball.....\$24.00
Howard the Duck.....\$32.00
Mindshadow.....\$32.00
Murder on the Mississippi.....\$32.00
Hacker II.....\$32.00

Addison-Wesley

Smart Eyes.....\$48.00
The Hobbit.....\$28.00
The Fellowship of the Ring.....\$32.00

Advanced Ideas

The Game Show.....\$32.00
Tic Tac Show.....\$32.00
Wizard of Words.....\$32.00
Language Arts Elementary Series
(Requires The Game Show).....\$39.75
Language Arts Grade 6
(Requires The Game Show).....\$16.00
Language Arts Grade 5
(Requires The Game Show).....\$16.00
Language Arts Grade 4
(Requires The Game Show).....\$16.00
Language Arts Grade 3
(Requires The Game Show).....\$16.00

American Education Computer

Spanish.....\$16.00
Science, Grades 3-4.....\$16.00
Science, Grades 5-6.....\$16.00
Science, Grades 7-8.....\$16.00
Spelling.....\$39.75
Phonics.....\$39.75
Learn to Read.....\$39.75
Reading Comprehension.....\$39.75
U.S. History.....\$16.00
U.S. Geography.....\$16.00

Artsci

Magic Window I/e.....\$106.00
Magic Window II.....\$106.00
MagicCalc.....\$71.00
Magic Office System.....\$176.00
Magic Window II extra (128K).....\$141.50

Ashton-Tate

DBase II (requires Z80).....\$415.00

Software cont.

Avalon Hill

Tournament Golf.....\$24.00
Gulf Strike.....\$24.00
Statis Pro Baseball.....\$28.00
Mission on Thunderhead.....\$20.00
Dreadnoughts.....\$24.00
Tac.....\$32.00
Under Fire.....\$47.50

Bantam Electronics Publishing

Card & Party Shop.....\$32.00
Comic Strip Maker.....\$32.00

Baudville

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

Beagle Bros

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

Bluechip Software

Millionaire.....\$39.75
The Challenge (Millionaire/Tycoon/
Baron/Teaching Guide).....\$79.50

BPI Systems

General Accounting.....\$279.00
Payroll.....\$279.00
Accounts Receivable.....\$279.00
Accounts Payable.....\$279.00
General Accounting (ProDOS).....\$300.00
Payroll (ProDOS).....\$300.00
Accounts Receivable (ProDOS).....\$300.00
Accounts Payable (ProDOS).....\$300.00
Inventory Control (ProDOS).....\$300.00

Broderbund Software

Science Tool Kit Master Module...\$55.50
Science Tool Kit Module 1:
Speed and Motion.....\$32.00
Science Tool Kit Module 2:
Earthquake Lab.....\$32.00
Where in the USA is
Carmen San Diego?.....\$36.00

Software cont.

Broderbund Software cont.

Where in the WORLD is
Carmen San Diego?.....\$32.00
Type.....\$36.00
On Balance.....\$79.50
Bank Street Writer 64K.....\$55.50
Bank Street Writer 128K.....\$55.50
Bank Street Mailer 64K.....\$55.50
Bank Street Mailer 128K.....\$55.50
Bank Street Filer 64K.....\$55.50
Bank Street Filer 128K.....\$55.50
Bank Street Speller.....\$55.50
Dazzle Draw.....\$47.50
The Print Shop.....\$39.75
Print Shop Companion.....\$32.00
Print Shop Graphics Library Disk 1...\$20.00
Print Shop Graphics Library Disk 2...\$20.00
Print Shop Graphics Library Disk 3...\$20.00
Animate.....\$55.50
The Toy Shop.....\$47.50
Variable Feasts.....\$39.75
Airheart.....\$28.00
Lode Runner.....\$28.00
Championship Lode Runner.....\$28.00
Breakers.....\$36.00
Karateka.....\$28.00
Captain Goodnight and the
Islands of Fear.....\$28.00

DLM Inc.

Alien Addition.....\$27.00
Verb Viper.....\$27.00
Spelling Wiz.....\$27.00
Number Farm.....\$24.00
Create With Garfield.....\$24.00
Create w/Garfield - Delux.....\$32.00
Teddy Bear-rels of Fun.....\$32.00
Fish Scales.....\$24.00
Meteor Multiplication.....\$27.00
Alligator Mix.....\$27.00
Decimal Discovery.....\$36.75
Writing Adventure.....\$37.00

EPYX

Temple of Aphsai Trilogy.....\$32.00
Ballblazer.....\$32.00
Rescue on Fractalus.....\$32.00
Championship Wrestling.....\$32.00
Destroyer.....\$32.00
The Movie Monster Game.....\$32.00
World Games.....\$32.00
Summer Games.....\$32.00
Summer Games II.....\$32.00
Winter Games.....\$32.00
World's Greatest Baseball Game...\$32.00
World's Greatest Football Game...\$32.00

Infocom

Zork I.....\$32.00
Zork II.....\$36.00
Zork III.....\$36.00
Deadline.....\$39.75
Wishbringer.....\$32.00
Suspended.....\$39.75
Planetfall.....\$32.00
The Witness.....\$32.00
Infidel.....\$36.00
Seastalker.....\$32.00
Cutthroats.....\$32.00
Suspect.....\$36.00
A Mind Forever Voyaging.....\$36.00
Enchanter Trilogy.....\$63.75
Enchanter.....\$32.00
Sorcerer.....\$36.00
Spellbreaker.....\$39.75
Trinity.....\$32.00
Fooblitzky.....\$32.00
Ballyhoo.....\$32.00
Hitchhiker's Guide.....\$32.00
Moonmist.....\$32.00
Starcross.....\$39.75

Micropro International

Wordstar Professional.....\$349.50
Wordstar.....\$247.00

Microprose Software

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

Quality Software

Bag of Tricks II.....\$39.75

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

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

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

What is a library disk?

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

Issue	Mag \$4.75	Disk \$9.95	Both \$12.95
42	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
41	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
39	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
38	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
37	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
36	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
35	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
34	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
33	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
31	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
29	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
28	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
27	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
22	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
17	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
..... Book of Softkeys Vol 3			
15	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
13	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
12	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
..... Book of Softkeys Vol 2			
10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
☆ 7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
..... Book of Softkeys Vol 1			
4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Core 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Core 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Core 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computing 3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Best of Hardcore Computing	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Core Special \$10.00 (All three CORE magazines)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Some disks apply to more than one issue and are shown as taller boxes. Special "Both" disk & magazine combination orders apply to one issue and its corresponding disk.
 Book Of Softkey titles are inserted for reader information only. Each book contains all of the softkeys appearing in the issues below the title line. Please consult current Book of Softkey ad for current prices and ordering information.
 ☆ We have a limited supply of these issues.
 ● Back issue is no longer available

Rates For Foreign Orders

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

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

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

 _____ Exp. _____

Signature _____ CP43

Send check or money order to: COMPUTIST PO Box 110846-T Tacoma, WA 98411.
 Most orders are shipped within 5 working days, however please allow 4-6 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. **U.S. funds drawn on U.S. bank only.**

Collection Completers!

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

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

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

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

Description of Available Back Issues.

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

39 *Sofkeys* | MIDI/8 Plus | Homeword v2.1 | Borrowed Time | Amazon | Speed Reader II | *Readers' Sofkeys* | Discovery! | M-ss-ng L-nks series | Donald Ducks's Playground | Mastering the SAT | Copy II 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 | Random | *Features* | Electronic Disk Drive Swapper | Abusing the Epilogues | Print Shop Companion's Driver Game | *Core* | Keyboard Repair | Fixing the Applesoft Sample Disk | *Hints* | Carmen Sandiego |

38 *Sofkeys* | Cyclod | Alternate Realty | Boulder Dash I & II | Hard Hat Mack (Revisited) | The Other Side | *Readers' Sofkeys* | 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 *Sofkeys* | Under Fire | Pegasus II | Take 1 (revisited) | Flight Simulator II v1.05 (part 2) | *Readers' Sofkeys* | 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 II Plus 6.0: a review | The DOS Alterer |

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

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

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

33 *Sofkeys* | Word Juggler | Tink! Tonk! | Sundog v2.0 | G.I. Joe & Lucas Film's Eidolon | Summer Games II | Thief | Instant Pascal | World's Greatest Football Game | *Readers' Sofkeys* | 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 *Sofkeys* | Revisiting Music Construction Set | Cubit | Baudville Software | Hartley Software | Bridge | Early Games for Young Children | Tawala's Last Redoubt | *Readers' Sofkeys* | Print Shop Companion | Cracking 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 *Sofkeys* | Trivia Fever | The Original Boston Computer Diet | Lifesaver | Synergistic Software | Blazing Paddles | Zardax | *Readers' Sofkeys* | 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 *Sofkeys* | Millionaire | SSI's RDOS | Fantavision | Spy vs. Spy | Dragonworld | *Readers' Sofkeys* | 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 *Sofkeys* | Threshold | Checkers v2.1 | Microtype | Gen. & Organic Chemistry Series | Uptown Trivia | Murder by the Dozen | *Readers' Sofkeys* | 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 *Sofkeys* | Ultima IV | Robot Odyssey | Rendezvous | Word Attack & Classmate | Three from Mindscape | Alphabetic Keyboarding | Hacker | Disk Director | Lode Runner | MIDI/4 | *Readers' Sofkeys* | 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 *Sofkeys* | Microzines 1-5 | Microzines 7-9 | Microzines (alternate method) | Phi Beta Filer | Sword of Kadash | *Readers' Sofkeys* | 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 *Sofkeys* | Cannonball Blitz | Instant Recall | Gessler Spanish Software | More Stickybears | *Readers' Sofkeys* | Financial Cookbook | Super Zaxxon | Wizardry | Preschool Fun | Holy Grail | Inca | 128K Zaxxon | *Feature* | ProEdit | *Core* | Games of 1985 part I |

25 *Sofkeys* | DB Master 4.2 | Business Writer | Barron's Computer SAT | Take 1 | Bank Street Speller | Where In The World Is Carmen Sandiego | Bank Street Writer 128K | Word Challenge | *Readers' Sofkeys* | 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 *Sofkeys* | Electronic Arts software | Grolier software | Xyphus | F-15 Strike Eagle | Injured Engine | *Readers' Sofkeys* | Mr. Robot And His Robot Factory | Applecillin II | Alphabet Zoo | Fathoms 40 | Story Maker | Early Games Matchmaker | Robots Of Dawn | *Feature* | Essential Data Duplicator copy parms | *Core* | Direct Sector Access From DOS | ..

22 *Sofkeys* | Miner 2049er | Lode Runner | A2-PB1 Pinball | *Readers' Sofkeys* | 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 *Sofkeys* | Sargon III | Wizardry: Proving Grounds of the Mad Overlord and Knight of Diamonds | *Reader' Sofkeys* | The Report Card V1.1 | Kidwriter | *Feature* | Apple II Boot ROM Disassembly | *Core* | The Graphic Grabber v3.0 | Copy II+ 5.0: A Review | The Know-Drive: A Hardware Evaluation | An Improved BASIC/Binary Combo |

19 *Readers' Sofkeys* | Rendezvous With Rama | Peachtree's Back To Basics Accounting System | HSD Statistics Series | Arithmetick | Arithmeticks 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 *Sofkey* | Sensible Speller for ProDOS | Sideways | *Readers' Sofkeys* | Rescue Raiders | Sheila | Basic Building Blocks | Artsci Programs | Crossfire | *Feature* | Secret Weapon: RAMcard | *Core* | The Controller Writer | A Fix For The Beyond Castle Wolfenstein Softkey | The Lone Catalog Arranger Part 1 |

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

1 *Sofkeys* | 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 |

Looking for the Best Deal in Town?

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

This package contains:

- ▶ **TWO DISKS** (supplied in DOS 3.3). Each containing at least 60 Super IOB Controllers including the standard, swap, newswap and fast controllers. In addition, each disk has the Csave program from COMPUTIST No. 13; version 1.5 of Super IOB; 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.**

Disk 1

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

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

Disk 2

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

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

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

*Requires at least 64K of memory.

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