

For The Serious User Of Apple][Computers

COMPUTIST

June 1986

Issue No. 32 \$3.75

Softkeys For:

Cubit

Bridge

Moebius

Early Games

Hartley Software

Baudville Software

Music Construction Set

5th Anniversary Issue

Feature:

Using Data Disks
with Microzines

Core:

Super IOB v1.5
a reprint



(Page 14)

COMPUTIST
PO Box 110937-A
Tacoma, WA 98411

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

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

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

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

Commands And Controls: In any article appearing in COMPUTIST, commands which a reader is required to perform are set apart from normal text by being indented and bold. An example is:

PR#6

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

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

6 P

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

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

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

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

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

Super IOB: This program has most recently appeared in COMPUTIST No. 32. Several softkey procedures will make use of a Super IOB controller, a small program that must be keyed into the middle of Super IOB. The controller changes Super IOB so that it can copy different disks. To get the latest version of this program, you may order COMPUTIST No. 32 as a back issue or order Program Library Disk No. 32.

RESET Into The Monitor: Some softkey procedures require that the user be able to enter the Apple's system monitor during the execution of a copy protected program. Check the following list to see what hardware you will need to obtain this ability.

Apple II Plus - Apple IIe - 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 II Plus - Apple compatibles: 1) Install an F8 ROM with a modified RESET vector on the computer's

motherboard as detailed in the "Modified ROM's" article of COMPUTIST No. 6 or the "Dual ROM's" article in COMPUTIST No. 19.

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

Recommended Literature: The Apple II 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, \$19.95; *Assembly Language For The Applesoft Programmer*, Roy Meyers and C.W. Finley, Addison Wesley, \$16.95; and *What's Where In The Apple*, William Lubert, Micro Ink., \$24.95.

Keying In Applesoft Programs: BASIC programs are printed in COMPUTIST in a format that is designed to minimize errors for readers who key in these programs. To understand this format, you must first understand the formatted LIST feature of Applesoft.

An illustration - If you strike these keys:

10 HOME:REMCLEAR SCREEN

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

10 HOME : REM CLEAR SCREEN

Programs don't usually LIST the same as they were keyed in because Applesoft inserts spaces into a program listing before and after every command word or mathematical operator. These spaces usually don't pose a problem except in line numbers which contain REM or DATA command words. The space inserted after these command words can be misleading. For example, if you want a program to have a list like this:

10 DATA 67,45,54,52

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

10 DATA 67,45,54,52

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

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

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

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

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

CALL -151

Now key in the hexdump exactly as it appears in the magazine ignoring the four-digit checksum at the end of each line (a "\$" and four digits). If you hear a beep,

you will know that you have typed something incorrectly and must retype that line.

When finished, return to BASIC with a:

E003G

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

Keying In Source Code The source code portion of a machine language program is provided only to better explain the program's operation. If you wish to key it in, you will need an assembler. The S-C Assembler is used to generate all source code printed in COMPUTIST. Without this assembler, you will have to translate pieces of the source code into something *your* assembler will understand. A table of S-C Assembler directives just for this purpose was printed in COMPUTIST No. 17. To translate source code, you will need to understand the directives of your assembler and convert the directives used in the source code listing to similar directives used by your assembler.

Computing Checksums Checksums are four digit hexadecimal numbers which verify whether or not you keyed a program exactly as it was printed in COMPUTIST. There are two types of checksums: one created by the CHECKBIN program (for machine language programs) and the other created by the CHECKSOFT program (for BASIC programs). Both programs appeared in COMPUTIST No. 1 and The Best of Hardcore Computing. An update to CHECKSOFT appeared in COMPUTIST No. 18. If the checksums these programs create on your computer match the checksums accompanying the program in the magazine, then you keyed in the program correctly. If not, the program is incorrect at the line where the first checksum differs.

1) To compute CHECKSOFT checksums:

**LOAD filename
BRUNCHECKSOFT**

Get the checksums with

&

And correct the program where the checksums differ.

2) To compute CHECKBIN checksums:

**CALL -151
BLOAD filename**

Install CHECKBIN at an out of the way place

BRUN CHECKBIN, A\$6000

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

xxx.xxx Y

And correct the lines at which the checksums differ.

Coping with COMPUTIST

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

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

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

ESQUIRE, January 1982: *Secrets of the Software Pirates* by Lee Gomes: When some Apple enthusiasts in Washington State heard about the boycott (of Locksmith advertisements), they decided that it was nothing but censorship and another example of the magazines' ignoring the average Apple user to placate their advertisers. So they started their own publication, **HARDCORE COMPUTING**, which with its first issue came to play the role of iconoclastic underground magazine battling the industry's stodgy straight press.

TIME, February 8, 1982: *Roaming Hi-Tech Pirates*: **HARDCORE COMPUTING**, a small magazine in Tacoma, Washington, warns pirates about the latest technology that companies are using against them.

5th Anniversary Issue

Yes, it has been five years since this magazine's inception: 5 years of rabid copy-protection by major software publishers, 5 years of censorship by major computer magazines about undoing copy-protection, 5 years of struggle and change for SoftKey Publishing in its effort to reveal suppressed information about copy-protection, 5 years of devoted support by our subscribers.

It all began in the attic of a small house in Tacoma, Washington (you know, that small 'town' next door to Seattle?) around midnight, the hacker's bewitching hour when everything is asleep except for the computers.

In the summer of '81, it seemed as if there was a plot to boycott a copy program called 'Locksmith' because, unlike Apple's COPYA, this program would copy disks that software publishers didn't want anyone, even the person who bought the disk, to backup.

Editors wrote editorials denouncing as pirates anyone who backed up their copy-protected software. Some even went so far as to distinguish between different sorts of pirates: commercial operators out to make a killing in violation of the US copyright laws, casual copiers out to expand their library by exchanging the copies they made of their own software, and accidental pirates who allow their software to be copied by one of the casual copiers.

Here are excerpts from some of those editorials:

"The presence of a number of commercially available bit copy programs will probably stimulate many software companies to review their service policies. Those reviews should produce constructive new policies which improve pricing and availability of replacement packages. That will be a positive step.

The presence of commercially available bit copy programs can make it easier for software pirates to make unauthorized copies of commercial software for broad distribution. That is potentially a negative result.

But the bit copier itself is not logically or legally responsible for either. It is simply a program — one which can be used badly or well. The simple fact is, bit copy programs are now here."

Mike Harvey, publisher-editor
NIBBLE, volume 2, No.2

"Several other magazines in the industry have recently been running advertisements for a program that copies (duplicates) protected software for a particular machine. One of the magazines (Micro) checked with us to see our



We are NOT

PIRATES

but we're not fools, either.

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

Remove copy-protection

Remove copy-protection from your valuable library of expensive software. The publisher of **COMPUTIST** has been showing subscribers how to unlock and modify commercial software for the past 5 years. Don't be one of the users abused by user-FRIENDLY locked-up software.

6 issue SUBSCRIPTION RATES: U.S.: \$20 First class: \$24 Canada, Mexico: \$34 Other Foreign: \$60

SAMPLE COPY: U.S.: \$4.75 Foreign: \$8.00

US funds drawn on U.S. bank. Send check or money order to:

COMPUTIST, PO Box 110937-A, Tacoma, WA 98411

NEW subscriber Renew my subscription New address

Name _____ D# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

 _____ Exp. _____

Signature _____ CP32

subscribers, renew your freedom

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

and if you're moving...

Let us know right away or at least 30 days in advance so that you won't miss a single issue. Just write your new address on this order form, and include your present address label. Issues missed due to non-receipt of this Change-of-Address may be acquired at the regular back-issue rates.

Please remember, the Post Office does not forward third class mail unless requested.

feelings on such advertising. We indicated that we wouldn't run such ads, given that the software in question could be used to produce copies of "protected" and proprietary software. We understand that Micro has since decided to do the same.

We applaud this move toward protecting the rights of the software industry and encourage additional comment."

Robert Lock, publisher-editor
COMPUTE, volume 3, No.3

"MICRO is unconditionally opposed to the illegal copying of software listings, cassettes, diskettes or any other protected material. I am embarrassed that the need has arisen to make what should be an obvious statement of policy. Due to the publication of an advertisement in our January and February issues, a number of advertisers and readers have expressed concern that MICRO appeared to be supporting the illegal copying of protected disks. In retrospect, I believe that I made a mistake in allowing the ad to run..."

"The advertisement referred to above was for

a program that would copy 'protected' diskettes.

The staff of MICRO will now review every ad and where there are problems, take appropriate steps."

Robert M. Tripp, publisher-editor
MICRO, No.34, March 1981

And so began the censorship of advertising which continues unabated to this very day, even though the bit copy programs are now advertised with the disclaimer that they should not be used to pirate software.

However, our own ads are still being censored or refused, usually because our ads say that we show readers **how to remove (softkey) copy-protection** from their disk software.

That is why we are sending this special ANNIVERSARY issue to potential subscribers throughout the year. We took the normal COMPUTIST issue just before it was to go to print, added this special anniversary note and inserted the Computer Learning Center's catalog of public domain software as part of our subscription drive.

new to hacking?

COMPUTIST magazine is dedicated to the *serious* user of Apple II computers. However, each issue contains informative articles dealing with subjects even novices can follow.

Our statistics show that once you've read this far, you're with us for life. As a matter of fact, most COMPUTIST readers prefer our magazine to any others they receive or have received (and the rest are pretty darn happy with it).

For You

Through the years, COMPUTIST has had two main goals: 1) to show our readers, in as clear a format as possible, how to copy the "uncopyable" programs they have purchased and 2) to teach our readers what they want to know about their computer.

Each month we detail the copying of several commercial programs. Whether an old classic like Miner 2049er (in COMPUTIST No. 27) or a new masterpiece like Fantavision (in COMPUTIST No. 30) you can be sure it has been or will be covered in COMPUTIST.

We listen to our readers' needs. If you have a program that you wish you could copy, just tell us and if we haven't already printed information about that program, we will add it to our "Most Wanted List," which is a list of the software you will most likely see covered in future issues of COMPUTIST.

At the Core

To add a little flavor to our publication, each issue contains a CORE section which can contain anything from high-quality free software to tutorials on a variety of subjects (computer hardware modifications, the ins and outs of DOS, etc).

Through the CORE section, you'll learn about about the VTOC, sector editing, modified ROMs, Ultima IV characters, boot code tracing and much much more. Our readers are our authors, so as they learn you will too. You may even wish to write an article yourself and earn extra money.

You will find that as COMPUTIST takes you by the hand through the dark recesses of the machine language Monitor you will learn more about how and why your computer does what it does than you could learn from reading a textbook.

Magazine Reference Material

Back issues of COMPUTIST are considered a valuable asset when you need hints on repairing damaged files or converting a disk you can't duplicate into one that you can. A survey we took of our readers showed that 91.2% of them keep every issue of COMPUTIST and add it to their reference material library. How many other magazines can you say that about?

COMPUTIST is Best

Thousands of satisfied readers can't be wrong. Whether it is removing copy protection from commercial disks or teaching about the heart of your computer, COMPUTIST is best!

We thank all our loyal subscribers and readers for their continuing support.

into the past...

Yes, we were once called *HARDCORE Computing*, then we became *CORE* and *Hardcore COMPUTIST*, and now simply *COMPUTIST*. For those of you who are confused, here's a list of all our publications to date.

1981

Hardcore Computing No. 1
Update 1.1 newsletter

1982

Hardcore Computing No. 2
Update 2.1 newsletter
Hardcore Computing No. 3
Update 3.1 newsletter
Update 3.2 newsletter

1983

CORE No. 1 Graphics special
Hardcore COMPUTIST 1
Hardcore COMPUTIST 2
CORE No. 2 Utilities special
Hardcore COMPUTIST 3
Hardcore COMPUTIST 4
CORE No. 3 Games special
Hardcore COMPUTIST 5

1984

Hardcore COMPUTIST 6
Hardcore COMPUTIST 7
Hardcore COMPUTIST 8
Hardcore COMPUTIST 9
Hardcore COMPUTIST 10
Hardcore COMPUTIST 11
Hardcore COMPUTIST 12
Hardcore COMPUTIST 13
Hardcore COMPUTIST 14
Hardcore COMPUTIST 15

1985

Hardcore COMPUTIST 16
Hardcore COMPUTIST 17
Hardcore COMPUTIST 18
Hardcore COMPUTIST 19
Hardcore COMPUTIST 20
Hardcore COMPUTIST 21
Hardcore COMPUTIST 22
Hardcore COMPUTIST 23
Hardcore COMPUTIST 24
Hardcore COMPUTIST 25
Hardcore COMPUTIST 26

1986

COMPUTIST 27
COMPUTIST 28
COMPUTIST 29
COMPUTIST 30
COMPUTIST 31
COMPUTIST 32

COMPUTIST IS NOW PUBLISHED 12 times a year. Some back issues are still available (see the back issues ad on the back page!)

CREATURE MAKER



USE UP &
DOWN TO
SELECT
PART. USE
LEFT & RIGHT
TO SELECT
CREATURE.

HEAD
RABBIT
BODY
FRANK
LEGS
CLOWN

PRESS <RETURN> WHEN DONE



SOCCER



This month's cover:

Graphics from Baudville's "Prince."

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 110937-A, Tacoma, WA 98411.

Return postage must accompany all manuscripts, drawings, photos, disks, or tapes if they are to be returned. Unsolicited manuscripts will be returned only if adequate return postage is included.

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.

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

SUBSCRIPTIONS: Rates (for 6 issues): U.S. \$20, U.S. 1st Class \$24, Canada & Mexico \$34, Foreign \$60. Direct inquiries to: COMPUTIST, Subscription Department, PO Box 110937-A, Tacoma, WA 98411. Please include address label with correspondence.

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

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

COMPUTIST

June 1986

Issue 32

Publisher/Editor: Charles R. Haight Managing Editor: Ray Darrah
Technical Editor: Robert Knowles Circulation: Michelle Frank, 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

softkeys:

10 Revisiting Music Construction Set

by Joseph Weeks

12 Cubit

by Panagiotis Simatos

14 Baudville Software

by William Forsyth

23 Hartley Software

by Dave Stanton

24 Bridge

by Dick Meikle

26 Early Games for Young Children

by Joe Drelling

28 Tawala's Last Redoubt

by Danny Pollak

feature:

25 Using Data Disks with Microzines

by Wesley R. Felty

core:

16 Super IOB v1.5: a reprint

This article presents the program used most frequently by COMPUTIST to deprotect commercial software. by Ray Darrah

departments:

4 Input

6 Most Wanted List

6 Bugs

28 Adventure Tips

7 Readers' Softkey & Copy Exchange

Broderbund's Print Shop Companion by Roger Sum-on & David Chun-wai, Pirates Harbor's Kracking Vol II by Doni G. Grande, Origin System's Moebius by Jeff Rivett, International Solutions' Mouse Budget, Mouse Word & Mouse Desk by Bill Scherer, Electronic Arts' Adventure Construction Set by Charles Taylor

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.

Various APTs

I saw Conan on your most wanted list and have an APT for it. At the beginning of the game climb up to the top of the castle and walk over to the right. Jump off of the castle into the tree. This will give you an extra life, but it will only work once.

Now here is a super APT for Ghostbusters. When the program asks you for your name type in GOO. When prompted for the account number, type in all the 1's you can until it stops. This should give you about \$2000 or more.

Here's one for Rescue Raiders. When you start the game, rest your helicopter on the pad. Then type in POPPY. This should give you some functions that are surprising. You will be able to use these commands now:

J = teleports you to the enemy's base
K = teleports to middle of battle field
L = teleports you to your side of the battle field
/ = gives you a free helicopter
RETURN = makes you bullet proof (it does not shield you against missiles)

If for some reason your code doesn't work try typing ZIPPY instead of POPPY.

Kenny Khoo
Riverside, CA

Uni-Appleworks

Here's a quick (?) and dirty fix to get Appleworks and Sensible Speller together on a Unidrive 3.5. AW version 1.2 will transfer and work fine, but will not support the new Apple memory card. Give your dealer 20 bucks for an update if you need this feature. Take a backup of Sensible Speller ProDOS (COMPUTIST No. 16) and use a block (or sector) editor to search the file SPELLER for the string /SENSIBLE001/SPELLER. In my version this is at the beginning of block \$4A. Change this to /APPLEWORKS/S/SPELLER.

Transfer all the files from the AW Startup and Program disks to the Unidrive volume /APPLEWORKS. Create a subdirectory /S on this volume and transfer the modified SPELLER file onto this subdirectory along with all dictionary files. RENAME the SPELLER file to SPELLER.

When you quit AW press <return> to accept current prefix. When you are prompted with ENTER PATHNAME OF NEXT APPLICATION type S/SPELLER and you're off and running.

P.S. I like to set up another subdirectory on the main volume called /AW.DATA and configure AW to use this as my primary data source. This keeps things neat and avoids the 51 file limit on main ProDOS directories.

David Todd
Cambridge, MD

Different Gato Movie Maker Temple of Apschai

First off, your magazine is great. In COMPUTIST No. 23, Clay Harrell wrote a softkey for Gato by Spectrum Holobyte. Step 2 of his softkey tells us to read in Track \$15, Sector \$0C. I did this and found that the bytes on my version were different than on his. I read in the rest of the sectors on track \$15 and this problem is easily fixed. Simply load in track \$15, Sector \$0A for you who have different versions. The complete track should read:

TRACK	SECTOR	BYTE	FROM	TO
\$15	\$0A	\$8C	\$AD	\$A9
\$15	\$0A	\$8D	\$E9	\$01
\$15	\$0A	\$8E	\$C0	\$D0
\$15	\$0A	\$8F	\$A9	\$87

I don't know if it's the same with all versions or if it was simply a typo, but it worked for me.

Also, the new Movie Maker published by Electronic Arts' new label system is very easily deprotected on my version. Please note that my friend tried this softkey on his version and option 2 from the menu did not work, but if you have the same one that I do, this will work.

Requirements:
Blank disk
Sector editor
Movie Maker (Electronic Arts)
COPYA

1) Copy the original with COPYA or any other fast copy program, ignoring any errors on track \$06.

2) Get out your sector editor, and perform the following sector edits:

TRACK	SECTOR	BYTE	FROM	TO
\$01	\$0F	\$00	\$4C	\$18
\$01	\$0F	\$01	\$69	\$60
\$01	\$0F	\$02	\$A0	\$DD

Also, one more quick antique softkey for the Temple of Apschai:

Requirements:
Blank disk
Temple of Apschai
Demuffin (Advanced Demuffin is preferred)
System Master

1) Using Demuffin, or I prefer Advanced Demuffin, demuffin tracks \$03-\$22 to your blank disk.

2) Now, run Master Create on your system master and follow the instructions copying a normal DOS onto your copy.

3) I suggest you use a fast DOS for your copy since there is a lot of disk activity.

That's it!

P.S. I have many cheats and maps for Ultima IV and Quest of the Avatar by Lord British. If you (or your readers) want further information, please contact me.

Greg Poulos
23278 Ennishore Dr.
Novi, MI 48050
(313) 348-0183

input

Senior PROM News

Cutting Edge Enterprises, a Detroit-based producer of software and hardware products for the Apple II series of computers, announced the release of Version 2.0 of its utility and programming aid, the Senior PROM. The Senior PROM is firmware mounted on a small printed circuitry board which replaces the systems ROMs in the Apple IIc and IIe, and provides a variety of utility programs for examinations of memory, disks, and for writing and debugging Assembly-Language programs.

This new release provides approximately twice as many functions as were previously available, including keys which display both the region of memory that was executing at the point of examination and the Stack area. The most unique feature of the new release is a program (in ROM) that is comparable to the "Switcher" program for the Macintosh; it allows the user to keep two 64K programs in memory and switch between them in a matter of seconds. A graphics program can be combined with a database, for example, or DOS 3.3 and ProDOS can be exchanged back and forth instantly.

Other features include full NMI support allowing a saved program to be restarted at any time (from either memory or a normal DOS disk), allowing flexibility in game-playing, adventuring, or execution of utility programs. Additional features include the "Absolute Reset", save all 64K of Main memory to Auxiliary memory, and full Step and Trace functions including an updated Mini-Assembler, with full support for the expanded 65C02 instruction set.

Also included are utilities for easy alteration of memory while a program is running, selection of the active bank in upper memory, and a printout of the text screen with a single keystroke. In addition, the sector editor (in ROM) also has its own operating system, so it may be used to edit and examine disks without booting DOS first. Extensive documentation is included with the product, including installation and operating instructions, with background material on the uses and applications of the Senior PROM. User support is available via a bulletin board running 300/1200 baud at 313-349-2954. The Senior PROM is available directly from Cutting Edge Enterprises (Box 43234 Ren Cen Station/Detroit, MI 48243) for \$79.95. Purchasers of earlier versions can upgrade to the new release for a fee of \$10.00.

Clay Harrell
Cutting Edge Enterprises

Ulysses and the Golden Softkey

I just got COMPUTIST No. 20 and was unable to copy my Wizardry with COPYA, as stated in step #1 of the Wizardry softkey. Can anyone help me?

After reading Denny Colt's softkey for Marauder in COMPUTIST No. 22, I sat down to have a go with Ulysses and the Golden Fleece. Here is my softkey.

Ulysses

Serra On-Line

Requirements:

One blank disk (both sides)
COPYA or similar copy program
Sector editor

- 1) Copy both sides of the disk with COPYA.
- 2) Use a sector editor to read in track \$17 sector \$C.
- 3) Change Byte \$2C from \$A8 to \$60 and write it back to the disk.
- 4) Read in track \$17, sector \$0E and change byte \$A4 from \$A8 to \$60.
- 5) Write it back to the disk and you're done. Side B does not need to be edited.

Robert M. Kesslick
Newport News, VA

Some IIc Notes

Thank you very much for the back issues which I ordered and just received, they were truly magnificent. However, there was a slight problem in the softkey for "Bank Street Writer IIc" on page 26, COMPUTIST No. 25. The problem occurred in step 5; when moving the page indicator that loads Boot1 to ROM. The computer I have is an Apple IIc and changing \$6659 to \$F0 brought the message "check disk drive" onto the screen. To overcome this problem I had to move page 08 to page \$98 and change the relative page \$08 locations to page \$98. This worked perfectly on my IIc.

Another problem I encountered was at step 17, I could not find \$A0 on Track 4, Sector 7, byte \$37. Instead, I had to make the following changes to track \$0D, Sector \$07 bytes \$7E through \$85 inclusive on the disk.

Old Value D4 D5 D6 D7 A5 96 BF 9A
New Value D5 AA AD DE D5 AA 96 DE

Incidentally, the version I have of BSW is "2-1".

Also, I have some good news for other Apple IIc users.

It is a well known fact that the Apple IIc is a closed case, and opening the case voids the warranty. However, there is still a method of adjusting the drive speed of the built in disk drive without opening the case. This ability is very handy when preserving a nibble count with EDD, Locksmith, or even with Nibbles Away III, which incidentally I feel is not that great.

If you look carefully at the bottom of the Apple IIc you will notice a few screws, and also another hole directly under the disk drive which is about 2 inches from the side of the computer. If you insert a small screwdriver into this hole and turn it slightly clockwise you will have increased the speed of the drive. By turning counter-clockwise you will decrease the speed.

So now when backing up a disk which has a nibble count track, instead of using someone else's disk drive, you will now be able to use your own. Also Copy Programs like EDD are well worth their cost now, because you can use mode #'s 4 and 6, which involve adjusting drive speed.

Do not forget to reset the drive speed back to normal!

P.S. I have been unable to Boot Code Trace as indicated on the letter on page 6 of COMPUTIST No. 26.

Could some of your readers elaborate on this please?

Imtiaz Haq
Doha, Qatar

Piracy Rebuttal Rebuttal

This concerns the letter from Donald Moses concerning The Black Rose. You don't understand! *Low Life Scum* like us have to pirate, or else my computer would be a \$2500 doorstop! Pirating gives us access to lots of programs that there is no way we could afford. No way! I use them for school reports (Printshop, Word Juggler) and I get good grades because my reports are neat. Some of my Commodore friends who have no pirating access spend all their money buying programs. I couldn't waste my money on over priced software!

Yes pirating is stealing, I couldn't agree more. But, I just could not afford *not* pirating. I feel guilty doing this white-collar

input

crime. How much could companies bring their price down by? A lot! A \$40.00 program will sell for \$20.00 dollars from Programs Plus. So don't tell me they're not overcharging us!

Well I'm sorry you feel the way you do, I feel you are missing a large software library (stolen, if you will) that could multiply your computer's capabilities! *But there is no way to afford them.*

A conservative pirate (and poor!)

The Chemist
Charlottesville, VA

A Happy Computist

I am happy to renew my subscription for an Apple magazine that pays for itself in just a "few miserable pages... of text" (as stated by George Blank in COMPUTIST No. 28). I left out the "hard to read" because I don't see how he can say this. The print is not smeared or smudged, the context is in a format that is easy to understand for both laymen and the long time programmer and you give short easy steps as well as expounding on the process. Hard to read? Definitely not! As for the "few miserable pages," yes, they are few but not at all miserable. How can they be? Why just one page of one issue can pay for the entire year's subscription.

When I heard of ACS (Adventure Construction Set) by Electronic Arts, I was so intrigued by the idea of making my own adventures that I purchased it as soon as it hit the market. Approximately 83 days later it "bombed" (two months prior to the softkey in COMPUTIST No. 24). Since the 90 day warranty wasn't quite up, it **only** cost me shipping and handling for a new copy, which came with a letter insinuating that the problem might be my disk drives. Five years of running other software without any problems at all made me doubt it was the drives. My mind was put to ease when I saw the softkey to this program (which worked flawlessly) and knew I never had to worry about this again.

In reply to George Blank's response in COMPUTIST No. 28, I would rather pay twenty dollars for a magazine that's 100% Apple than one that's only 16% Apple and if George writes his "ingredients" as other manufacturers (listing highest quantity to lowest) we see he puts ads and general articles before any computer material. I wouldn't worry much about not getting his support, he's probably afraid of losing customers to a real Apple magazine, and CALL-APPLE?

Personally, I never really cared for theirs when it first came out and I haven't looked back since.

One final note. I do see how George can say that casual copying hurts profits. The company doesn't get any money (\$5.00 - \$50.00) for repairs/replacements to those "bombed" disks after the 90 day warranty because we don't use the original. I understand the software companies say that we are not buying products, but only leasing them. All I can say to that is, "when that gets enforced, they've lost my business altogether."

David L. Goforth
Indian Springs, NV

Mr. Goforth: The advertisement for "The Best of Hardcore Computing" on page 30 of COMPUTIST No. 28 which contains the George Blank response you refer to is a reprint of the last page of Hardcore Computing Update 1.1 which was published in the fall of 1981.

This ad was a direct result of somebody's joy that Creative Computing has "bit the dust." You see, for a while this magazine would refuse to run our ads in a desperate attempt to keep the general public in the dark about copy protection (and deprotection). Only later in time after copy protection information slowly leaked out (through Hardcore Computing and other pioneers) did they finally submit to the awful "disgrace" of publishing Hardcore ads.

As it has done so many times in the past, the course of time has only allowed the survival of the fittest. Too bad for Creative Computing. Maybe next time.

bugs

COMPUTIST No. 28:

Softkey for Ultima IV:

The "POKE 47829,13" in line 1060 should be a "POKE 47829,213."

COMPUTIST No. 31:

Black Box:

The BASIC program listing has two errors. Substitute the following lines for the corresponding ones appearing in COMPUTIST No. 31:

```
360 GOSUB 780: IF PX=2 THEN PX=0: PY=2 * (PY=0)+11 * (PY=13): GOTO 420
```

```
750 HOME: TEXT: VTAB 11: PRINT "YOUR^ SCORE^ IS^ " INT (S-10 * ((M-8) / NA)) "%"
```

Most Wanted List

Need help backing-up a particularly stubborn program?

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

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

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

- Mouse Calc Apple Computer
- Apple Business Graphics Apple Computer
- Jane Arktronics
- Vishlend Microlab
- Catalyst Quark, Inc.
- Gutenberg Jr. & Sr. Micromation LTD
- Prime Plotter Primesoft Corp.
- The Handlers Silicon Valley Systems
- The Apple's Core: Parts 1-3 The Professor
- Fun Bunch Unicorn
- Willy Byte ... Data Trek
- Terrapin Logo V2.00 Terrapin Software
- Conan Datasoft
- Cycloid Sirius Software
- Adventure Microsoft
- Cranston Manor Sierra On-Line
- Snoggle Bröderbund
- Robot War Muse
- ABM Muse
- Mychess II Dalamost
- E-Z Learner Silicon Valley Systems
- Story Tree Scholastic
- Agent U.S.A. Scholastic
- Handicapping System Sports Judge
- Dollars & Sense Monogram
- Echo Plus Agranat Systries
- Great Cross Country Road Race Activision
- Raster Blaster Budge Inc.
- GATO v1.3 Spectrum Holohyte
- Odin Odesta

readers' softkey & copy exchange

Roger Sum-on & David Chun-wai's...

Softkey for Print Shop Companion

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

Requirements:

64K II Plus, IIe or IIc
Super IOB 1.5
Blank disk

The Print Shop is a powerful printer program. It can be used to make signs and greeting cards to send to other people. It can also let you create your own graphics for the cards, but you cannot edit the fonts or the borders. The Print Shop Companion is an addition to The Print Shop that lets you do that and much more with an enhanced graphic editor, font and border editors, and a creature maker (mix heads, bodies and legs of various characters).

The Print Shop Companion is similar to the Print Shop, in that it allows you to make a backup (only one). But when I tried to use the original Companion to update the softkeyed Print Shop (COMPUTIST No. 17, page 13), the computer hung up. I found that the program is looking for the protection on track \$22. So, I tried to fix it.

When looked at The Print Shop Companion with a nibble viewer, I found that the protection is similar to the Print Shop's. Tracks 0-\$21 are in a normal format and track \$22 is in a very strange format. This track is checked by (what else?) a nibble count routine.

After boot code tracing the program, I found that the start of the nibble count routine is at \$B619, just like Print Shop. It fills page \$BB00 with \$FFs and reads the disk for four markers (\$D4, \$D5, \$DE, \$DF) and read a pair of 4+4 encoded bytes. Then it jumps to \$B6B5 to read two more markers (\$F5, \$FA) and follows a process similar to Print Shop's to check the bytes on the disk.

First time I tried to disable it by simply putting an RTS (\$60) at \$B619 (on my PSC disk it's at track 3, sector 6, byte \$19). Look for the sequence \$A0 00 A9 FF, which is LDY #00, LDA #FF.

It booted up without checking the nibble count, but when I used it to print a calendar, it just printed a lot of rubbish. I figured this was because of some checksum on the nibble routine. To make the checksum come out all right, I added the original two bytes at \$B619-\$B61A together and came up with a sum

CREATURE MAKER



USE UP & DOWN TO SELECT PART. USE LEFT & RIGHT TO SELECT CREATURE.

HEAD

RABBIT

BODY

FRANK

LEGS

CLOWN

PRESS <RETURN> WHEN DONE

of \$A0. So I put a \$40 at \$B61A to go with the \$60 at \$B619 (\$60 + \$40 = \$A0) and The Print Shop Companion worked!

Now that the softkey for The Print Shop Companion is finished, the built-in copy program must be removed because it is no longer needed and the system will hang up if you press the Escape key while booting. The Companion has two checks for the Escape key. One is at track \$0, sector \$4, bytes \$FC & \$FD, and the other is at track \$B, sector \$8, bytes \$B0 & \$B1. Replace the \$C9 9B (CMP #S9B) with \$A9 01 (LDA #01) and the Escape key will be ignored.

To deprotect the disk, just use the controller below in Super IOB 1.5 and RUN it. The controller will make the necessary sector edits. Now you should have a COPYable version of The Print Shop Companion.

controller

```
1000 REM P.S. COMPANION CONTROLLER
1010 TK=0 : LT=34 : ST=15 : LS=15 : CD=WR : FAST=1
1020 GOSUB 490 : RESTORE : GOSUB 610
1030 GOSUB 490 : T1=TK : TK=PEEK (TRK) - 1 : RESTORE : GOSUB 310 : TK=T1 : GOSUB 610
1040 IF PEEK (TRK) = LT THEN 1060
1050 TK=PEEK (TRK) : ST=PEEK (SCT) : GOTO 1020
1060 HOME : PRINT "COPY" "DONE" : END
```

```
5000 DATA 6^ CHANGES
5010 DATA 0 , 4 , 252 , 169
5020 DATA 0 , 4 , 253 , 1
5030 DATA 3 , 6 , 25 , 96
5040 DATA 3 , 6 , 26 , 64
5050 DATA 11 , 8 , 176 , 169
5060 DATA 11 , 8 , 177 , 1
```

controller checksums

1000 - \$356B	5000 - \$F62B
1010 - \$0715	5010 - \$F04A
1020 - \$0A5C	5020 - \$7663
1030 - \$C4DB	5030 - \$6001
1040 - \$F817	5040 - \$1F20
1050 - \$2A47	5050 - \$82B9
1060 - \$0FC6	5060 - \$56A7

CALENDAR: CENTER

HIGHLIGHT SPECIAL OCCASIONS

MARCH, 1986
S M T W T F S

						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					


 MOVE CURSOR TO SELECT DATE
 PRESS <RETURN> TO ENTER TEXT

<ESC> TO GO BACK

readers' softkey & copy exchange

Doni G. Grande's softkey for...

Cracking Vol II

Pirates Harbor
POB 8928
Boston, MA 02114

Requirements:

Apple II or better
Super IOB v1.5
Sector editor

Pirates Harbor puts out some really good cracking tips disks and the ones by Disk Jockey are about the best. Strangely enough, the newest one is copy protected. However, if you have fully digested the information given in earlier volumes, this disk is very easy to copy.

Examination of the disk shows that Track \$00, Sectors \$00-\$09 and Track \$11, Sectors \$00-\$01 are in normal DOS format whereas the rest of the disk has altered data prolog marks of D4 AB AD instead of D5 AA AD. Also, the disk does a checksum to see if anything has been changed and hangs if so. The utility "Snoopy and the Doctor" included on the disk re-patches the data marks when it is executed, so this must be defeated too.

Step by Step

1) Install the controller with this softkey in Super IOB v1.5 and copy the original disk. This controller is a good example of how to copy partial tracks with Super IOB v1.5.

2) Make the following changes using a sector editor.

Track	Sector	Byte	From	To
\$00	\$02	\$53	\$D4	\$D5
\$00	\$02	\$58	\$AB	\$AA
\$00	\$02	\$E7	\$D4	\$D5
\$00	\$02	\$F1	\$AB	\$AA
\$00	\$0F	\$C2	\$4C	\$60
\$06	\$07	\$D5	\$D4	\$D5
\$06	\$07	\$DD	\$AB	\$AA

controller

```
1000 REM CRACKING BY THE DISK JOCKEY II
      CONTROLLER
1010 TK=0 : LT=1 : ST=9 : LS=15 : CD=WR : GOSUB
      1100 : FAST=1 : REM T$00, S$0-9
1015 RESTORE : GOSUB 210
1020 TK=0 : LT=0 : ST=15 : LS=9 : CD=WR : GOSUB
      1100 : REM T$00, S$0A-$0F
```

```
1022 TK=1 : LT=17 : ST=15 : LS=15 : CD=WR :
      GOSUB 1100 : REM T$01, S$00 TO T$10, S$0F
1025 GOSUB 230
1030 TK=17 : LT=18 : ST=1 : LS=15 : CD=WR :
      GOSUB 1100 : REM T$11, S$00-$01
1035 RESTORE : GOSUB 210
1040 TK=17 : LT=17 : ST=15 : LS=1 : CD=WR :
      GOSUB 1100 : REM T$11, S$02-$0F
1042 TK=18 : LT=35 : ST=15 : LS=15 : CD=WR :
      GOSUB 1100 : REM T$12, S$00 TO T$22, S$0F
1045 GOSUB 230
1050 HOME : PRINT "COPY^ DONE" : END
1100 REM READ/WRITE SUBROUTINE
1120 GOSUB 490 : GOSUB 610
1130 GOSUB 490 : GOSUB 610 : IF PEEK (TRK) = LT
      THEN 1150
1140 TK=PEEK (TRK) : ST=PEEK (SCT) : GOTO 1120
1150 RETURN
5000 DATA 212, 171, 173
```

controller checksums

1000	- \$356B	1042	- \$1731
1010	- \$E7DA	1045	- \$96D6
1015	- \$4ABB	1050	- \$0211
1020	- \$8503	1100	- \$CD4B
1022	- \$1804	1120	- \$E65A
1025	- \$49CD	1130	- \$75F4
1030	- \$7D89	1140	- \$A011
1035	- \$BFC9	1150	- \$169D
1040	- \$280F	5000	- \$54DD

Jeff Rivett's softkey for...

Moebius

Lord British/Origin systems
340 Harvey Road
Manchester, NH 03103

Requirements:

48K Apple or compatible
MOEBIUS original disk
A blank disk
Super IOB v1.5

Tricks of the trade #542 - if you can deprotect a disk produced by a certain company, and you have another disk by the same company that you want to deprotect, try the method you used on the first one before you do anything else.

I was very impressed with Mike Roetman's article on ULTIMA IV in COMPUTIST No. 28. The translate table protection is one I had never heard of. Should the POKE 47829,13 in line 1060 be POKE 47829,213? I did it the second way and it worked.

MOEBIUS also happens to be by Origin systems, so I tried the ULTIMA softkey on it.

The IOB controller worked like a charm, producing a DOS 3.3 formatted disk. As in the softkey for ULTIMA, however, some further modifications were needed before the copy would run.

I assumed that since the protection scheme had been identical up to this point, the modifications would at least be similar to those in the ULTIMA softkey. The ULTIMA mods involved changing two STA instructions to all NOP's, and swapping a \$B7 with a \$E8.

I searched the disk for the two STA instructions (8D5DBD and 8D2CBF), and found them in track \$04, sector \$07 and track \$03, sector \$00. I looked at the code in those sectors and came up with the changes listed below.

Step by Step

- 1) Boot a disk with your favourite DOS.
- 2) From BASIC, enter the following line. This lets DOS BRUN the boot program.

POKE 40514,52

- 3) Initialize a disk with the following line:

INIT B OOT

- 4) Run IOB with the controller for ULTIMA IV, using the disk you made in step 3 as the destination.

- 5) Make the changes in Table 1 below. If the value at one of the locations is already the same as the "To" value, leave it alone.

Table 1:

Track	Sector	Byte	From	To
03	00	37	E8	B7
		39	B7	E8
		48	8D	EA
		49	5D	EA
		4A	BD	FA
		4D	8D	EA
		4E	2C	EA
		4F	BF	EA
		56	8D	EA
04	07	57	5D	EA
		58	BD	EA
		5B	8D	EA
		5C	2C	EA
		5D	BF	EA
		5F	E8	B7
		61	B7	E8
		B5	8D	EA
		B6	5D	EA
		B7	BD	EA
04	07	BA	8D	EA
		BB	2C	EA
		BC	BF	EA
		BE	E8	B7
		C0	B7	E8

readers' softkey & copy exchange

Bill Scherer's softkey for...

Mouse Budget Mouse Word Mouse Desk

International Solutions, Inc.
910 W. Maude Ave.
Sunnyvale, CA 94086
(408)-773-0443
Mouse Budget: \$69.95
Mouse Word: \$129.95
Mouse Desk: \$39.95

Requirements:

COPYA
A sector editor
Blank disks
//e or //c with 65C02
A Mouse

Mouse Budget is a mouse-based personal budget program. It incorporates the pull-down windows and menus similar to the Macintosh. The program tracks deposits and expenditures by allowing you to define up to 60 separate accounts. Any account can be instantly viewed in windows on your screen. Mouse Budget also allows you to see your total net worth or yearly budget for any category.

The protection is easily defeated by first making a backup of the original with COPYA. Then use a sector editor to make the following change.

MOUSE BUDGET:				
Track	Sector	Byte	From	To
\$13	\$0C	\$44	00	10
		\$90	A0	60

Mouse Word is a mouse based word processor with sophisticated text formatting performed by using the mouse, a mail merge feature, an on-screen calculator, and a communications function for use with a modem.

The protection is easily defeated by first making a backup of the original with COPYA. Then use a sector editor to make the following change.

MOUSE WORD:				
Track	Sector	Byte	From	To
\$0F	\$0D	\$0A	EC	37

Mouse Desk is a desktop organizer and program selector that uses the mouse and pull-

down windows. Mouse Desk makes the Apple // series similar to the Macintosh where you can point and click the mouse to easily move, copy, and delete files on the disk. You can also switch back and forth between programs without rebooting.

The protection is easily defeated by first making a backup of the original with COPYA. Then use a sector editor to make the following change.

MOUSE DESK:				
Track	Sector	Byte	From	To
\$0C	\$04	\$54	D0	F0

Charles Taylor's softkey for...

Adventure Construction Set

Electronic Arts
2755 Campus Dr.
San Mateo, CA 94403

Requirements:

Copy program that can ignore errors
Sector editor
Four blank disk sides

After trying unsuccessfully to copy Adventure Construction Set using Steve and Rod Smith's method in COMPUTIST No. 24 (Page 10), I realized that I had a different version. In addition to not working after applying the softkey, my version gave a copy error only on track 6, not tracks 5 and 6 as described in the article.

I successfully copied my Adventure Construction Set by studying the Archon softkey (same article). I took my non-working copy and searched it for the byte sequence of \$4C 69 xx (as in JMP \$xx69). I found it in three places, twice on track 1, sector F (4C 69 A0) and once on track 9, sector B (4C 69 4D). After studying the code, I decided that the code on track 9 did not involve copy protection. I changed the two occurrences of 4C 69 A0 to 18 60 DD as suggested, wrote them back to my copy, and I had a working back up of Adventure Construction Set!

Be sure to disregard the ASCII string "DON'T BREAK THIS GAME" found on track 1, sector \$E.

Summary

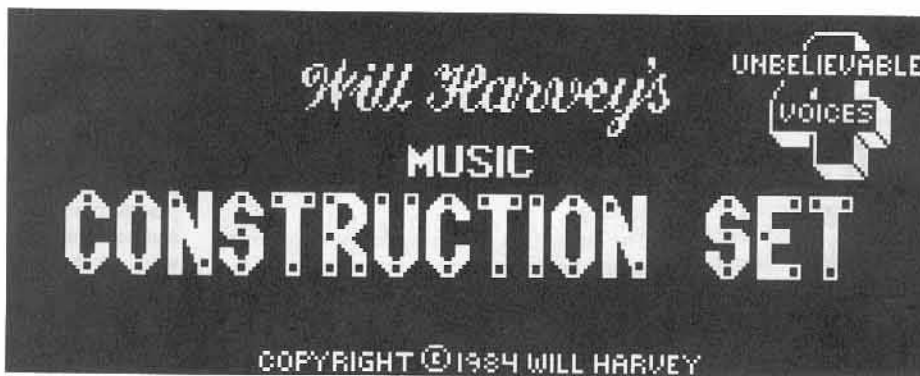
- 1) Copy the label side of the program disk with a program that will ignore the copy error on track 6.
- 2) Using a sector editor, make the following changes and write them back to your copy:

Track	Sector	Byte	From	To
\$01	\$0E	\$47	\$20	\$18
\$01	\$0E	\$48	\$F8	\$60
\$01	\$0E	\$49	\$A0	\$80
\$01	\$0E	\$4A	\$B0	\$70
\$01	\$0F	\$00	\$4C	\$18
\$01	\$0F	\$01	\$69	\$60
\$01	\$0F	\$02	\$A0	\$DD
\$01	\$0F	\$6F	\$4C	\$18
\$01	\$0F	\$70	\$69	\$60
\$01	\$0F	\$71	\$A0	\$DD
\$07	\$0F	\$52	\$53	\$74
\$07	\$0F	\$56	\$25	\$04

- 3) Copy the other three sides with a normal copier.



Revisiting the softkey for...



by Joseph Weeks

*Electronic Arts
2755 Campus Dr.
San Mateo, CA 94403*

Requirements:

48K Apple II and up
A blank disk
Super IOB

When I purchase a piece of software, I first test it, then I embark on the task of backing it up. Usually I am fortunate to find a softkey in COMPUTIST which will do the job for me.

I found Music Construction Set to be an excellent program that becomes a fantastic program if you have a Mockingboard. I checked COMPUTIST and found two Softkeys for MCS, one in COMPUTIST No. 9, page 7 and one in COMPUTIST No. 12, page 27, which even had a key for two versions of MCS.

Much to my dismay it soon became apparent that Electronic Arts had totally revised their copy protection on MCS and neither softkey would come close to cracking my version. This forced me into the enjoyable but sometimes frustrating task of boot code tracing.

The Protection

The protection scheme being used is one that seems to be quite popular in newer software, because it is uncopyable (see COMPUTIST No. 24, page 21). The disk is normal DOS 3.3 and can be copied with COPYA except for track 6. The reason for the unreadability of track 6 is that using very accurate recording equipment, tracks 5 and 5.5 were written identically. The data on track 6 is actually "track bleeding" from track 5.5 and is therefore unreadable. If you try to copy the disk with a nibble copier,

the Apple drive can't reliably copy both track 5 and 5.5, and therefore the copy will fail the protection check and will not boot all the way.

The Boot Code Trace

Although boot code tracing isn't required to perform the softkey, it was required for me to determine how, where, and what code needed to be modified to bypass the protection. If you do not wish to learn anything about boot code tracing this disk, skip straight to the end and copy it with Super IOB. However, you should read on. You might learn something.

Following my standard procedure for boot code tracing, we first write a known value to all user memory. I usually use \$11 or \$00.

CALL-151
800:00
801<800.BFFFM

Although the clearing of memory to zeros isn't necessary in this case, it always provides a method of looking through memory to see what has happened if you can't figure it out by disassembling the code. Next we move the boot code from ROM to RAM, change its exit to JuMP to the Monitor rather than the next stage, then execute the boot code. Note: this assumes an Apple Disk II controller in slot 6.

9600<C600.C6FFM
96F9:59 FF
9600G

The drive will come on and shortly you will get a beep as you jump to the monitor. Turn off the drive and see what we have.

C0E8
801L

This is what we get:

```
801- 4C 04 08 JMP $0804
804- A9 00 LDA #$00 ;Set the Reset
806- 8D F2 03 STA $03F2 ;vector to
809- A9 C6 LDA #$C6 ;$C600
80B- 8D F3 03 STA $03F3
80E- 49 A5 EOR #$A5
810- 8D F4 03 STA $03F4
813- A9 00 LDA #$00 ;Put $00
815- 8D C7 09 STA $09C7 ; at $9C7
```

```
818- AD E9 C0 LDA $C0E9 ;Turn drive on
81B- A9 B7 LDA #$B7
81D- 48 PHA ;Push a $B7
81E- A9 B0 LDA #$B0 ;Store a $B0
820- 85 3E STA $3E ; at $3E
822- A9 00 LDA #$00
824- 20 00 0C JSR $0C00 ;Load track 0
827- A9 FF LDA #$FF
829- 48 PHA ;Push an $FF
82A- A9 00 LDA #$00
82C- AD C7 09 LDA $09C7
82F- D0 D3 BNE $0804
831- 4C 05 A0 JMP $0A05 ; (jump thru RTS)
```

Looking at location \$800 we find an \$05 meaning the boot code has brought five sectors of track 0 into memory from \$0800 to \$0CFF. The code starting at \$801 first sets up the RESET vector to \$C600 then turns the drive on. Next a \$B7 is pushed onto the stack, a \$B0 is stored at \$3E, and a Jump SubRoutine sends us to \$0C00. The code at \$C00 will eventually load the entire track 0 into memory from \$B000 to \$BFFF then return. Upon returning, an \$FF is pushed onto the stack and then we JuMP to \$0A05. This is the hidden exit from this stage of the boot to the next stage. The entry point to the next stage is on the stack. The code at \$A05 is only there for confusion and the only important part of it is the RTS at the end. Since we JuMPed and not JSRed to \$0A05, when the RTS is executed, we will effectively jump to the last two bytes on the stack (\$B7FF) plus one or \$B800. Very clever, but once found easily defeated.

831:59 FF
801G

Normally you can't directly execute each stage of a boot, but because each stage of MCS has code within it to turn on the drive, it is possible. The drive will turn and then we get a beep as we again jump to the monitor. Turn off the drive and see what we got.

C0E8
B800L

```

B800- A0 00 LDY #500
B802- 84 4C STY $4C
B804- A9 20 LDA #20
B806- 85 4D STA $4D
B808- A9 FF LDA #FF
B80A- 91 4C STA ($4C),Y
B80C- C8 INY
B80D- D0 FB BNE $B80A
B80F- E6 4D INC $4D
B811- 10 F7 BPL $B80A
B813- A2 30 LDX #30
B815- 20 BB B9 JSR $B9BB

```

This code begins by putting \$FF's from \$2000 to \$7FFF, then the X is loaded with \$30 followed by a Jump SubRoutine to \$B9BB. The code following this seems to be invalid so \$B9BB must be where it all happens.

You can list and try to follow the code at \$B9BB if you wish, but I'll be honest and say I got lost fast. There are loops and look-up tables and who knows what else. At this point I became just a little frustrated and so in hope of finding something recognizable, I listed the entire code from \$B000 to \$BFFF.

The code from \$B500 to B6FF seemed very suspicious. At \$B503 was a routine which does direct disk access, looks for a normal DOS 3.3 data header, (\$D5AAD), then looks for a bunch of \$B4's. I assumed this to be the protection check because an earlier examination of track 5 and 5.5 with a nibble editor presented a bunch of \$B4's. So now the task again appeared simple, find the call to this routine and defeat it.

I decided that an RTS placed at location \$B660 should do the trick, but when any byte from \$B500 to \$B6FF was changed, the system would become confused. Electronic Arts is evidently doing a checksum on these pages of memory, to discourage such attempts as ours. This probably takes place in the code I got lost in.

Checksum routines usually count bits. I wanted to change the byte at \$B660 from a \$20 to a \$60, which is adding \$40 to the sum in that page of memory. If I subtracted \$40 bits elsewhere from that page it should balance and work. Right? Can't hurt to try.

```

B660:60
B6FF:82
B800G

```

It BOOTS! Changing two bytes will remove the boot protection on the MCS disk. The demo will play, but when option selection is finished we get the message: INSERT MCS DISK AND PRESS A KEY. Curses! Another protection check to be defeated.

I opened the disk drive door and pressed RESET twice, then entered the monitor and started printing all valid looking code from \$800 to \$BFFF. Looking through this I found what seemed to be code identical to that found at \$B500 to \$B6FF in the second boot stage. However, this code was from \$4C00 to \$4DFF. I then searched the disk for this code but was unable to find it. It must either be self-writing or encrypted on the disk, probably the latter.

Not having given up yet, I then looked for the calls to these protection check routines and

found them along with some other protection related routines from \$A39C to \$A7FF. Fortunately I was also able to find this code on the disk on track \$C, sectors 0, 2, and 4. Armed with their locations it is easy to change or bypass them.

On my disk the edits required were as follows:

Track	Sector	Byte	From	To
0	C	60	20	60
0	C	FF	C2	82
C	0	EC	20	18
C	0	ED	00	60
C	2	77	CE	60

Step by Step

- 1) Install the controller program at the end of this article into Super IOB v1.5.
- 2) Copy the Music Construction Set disk with Super IOB, saying yes to the format option.

More Stuff

Later, I purchased a newer version of Music Construction Set with a new option for selecting a printer interface. I tried this same controller on the new disk to make a backup. The copy booted fine and would run the demo but would either bomb into the monitor or tell me to insert the original MCS disk if I tried to start the program.

I checked and found the code which was added to provide the new printer interface option had caused the location of the JSR \$4C00 that went to check for the original disk to move. On my first version it was at \$A7EC (track C, sector 0, byte \$EC), but on the new version it was at \$A836 (track D, sector F, byte \$36). When properly patched it gives a working backup.

The following sector edits should replace the third and fourth sector edits in the Super IOB controller and the table above if your version of MCS asks for the type of printer interface.

Track	Sector	Byte	From	To
D	F	36	4C	18
D	F	37	00	60

In the controller, substitute these DATA statements.

```

5030 DATA 13, 15, 54, 24
5040 DATA 13, 15, 55, 96

```

There are probably more than just these two versions of MCS around (or are coming). If these edits fail, try just searching for JSR \$4C00 (20 00 4C) and replacing it with a CLC, RTS (18 60).

controller

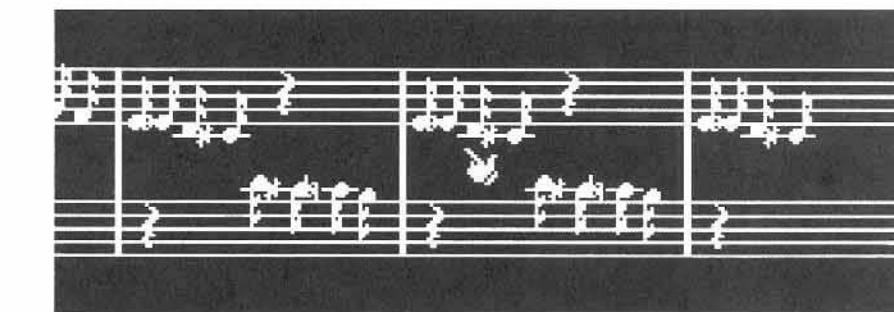
```

1000 REM MCS CONTROLLER
1010 ST = 15 : LS = 15 : FAST = 1 : CD = WR
1020 TK = 0 : LT = 6 : GOSUB 1050
1030 TK = 7 : LT = 35 : GOSUB 1050
1040 HOME : PRINT "COPY^ DONE" : END
1050 GOSUB 490 : GOSUB 610
1060 T1 = TK : TK = PEEK (TRK) - 1 : RESTORE : GOSUB
310 : TK = T1
1070 GOSUB 490 : GOSUB 610 : IF PEEK (TRK) = LT
THEN RETURN
1080 TK = PEEK (TRK) : ST = PEEK (SCT) : GOTO 1050
5000 DATA 5^ CHANGES
5010 DATA 0 ,12 ,96 ,96
5020 DATA 0 ,12 ,255 ,130
5030 DATA 12 ,0 ,236 ,24
5040 DATA 12 ,0 ,237 ,96
5050 DATA 12 ,2 ,119 ,24

```

controller checksums

1000	- \$356B	1080	- \$E172
1010	- \$16BA	5000	- \$73AE
1020	- \$6362	5010	- \$A22A
1030	- \$2E19	5020	- \$0D37
1040	- \$D5A4	5030	- \$D6EE
1050	- \$6D85	5040	- \$ADFC
1060	- \$745B	5050	- \$8DCA
1070	- \$7F39		



CUBIT

by Panagiotis Simatos

*Micromax Systems, Inc.
6868 Nancy Ridge Drive
San Diego, CA 92121*

Requirements:

Apple II series computer
A way to Reset into the monitor, or
A plug-in language card.

This is a softkey for the game Cubit (distributed by Micromax), with which I would like to emphasize on how to "grab" the game from memory using a language card, or any other memory card, in case you have no other way of resetting into the monitor.

If you are an experienced unprotector, you may skip most of the following and go directly to the step-by-step "grab" procedure. However, I would suggest that you hang on, since you may find some of the ideas that follow useful.

Whenever I decide to remove the copy protection from a game disk, I start paying attention to the following:

- The booting sounds, in connection with the booting time and whatever appears on the screen.
- The disk access during play, if any.
- The effect of RESET during play.

Now let's see how these apply to Cubit. First of all, it takes a long time to boot, which means that it is not a "quick-loader". By this term we mean disks with a special sectorless format, which load at lightning speed. The DOS on these disks is severely altered. This is certainly not the case with Cubit. You should also notice an extra "click" during the start of booting.

This is the disk arm recalibrating to track 0. The only thing that appears on the screen during booting is the Applesoft prompt.

From the above we can easily reach the conclusion that the DOS format of the disk has not been altered too much. With this in mind we go on.

Now it is time to examine disk access during play. In this case an experienced ear should easily realize that nothing is really loaded from the disk between rounds. The disk access that occurs seems to be a routine check of the program to see if the original disk is still in the drive. Keep this in mind, since this routine has to be defeated at a later stage.

What about the Reset key? If you try to press Reset during play, you will find that the program starts all over again. This is a feature that annoys me extremely, since the only way to quit the game is to turn the power off. However, speaking of unprotection, this feature is a blessing, since it reveals that the Reset vector (memory locations \$3F2 and \$3F3) holds the program start. Be careful not to confuse the program start with the code start. They may or may not be the same.

It seems that the normal FF sync bytes have been changed to a couple of different sequences, one preceding the the address field header and one preceding the data field header. Furthermore, the address and data field end marks have been changed to FF FF.

Anyway, a good bit copier, like Locksmith or Copy II Plus, should be able to recognize these alterations. You should copy tracks \$00 to \$22 in normal mode. However, in order to come up with a bootable copy, you should also do the following:

1) Recopy track \$00 synchronized (remember that extra arm recalibration to track \$00 during booting?)

2) Write-protect your copy before booting. This applies to a considerable number of commercial programs.

From now on you should set aside your original disk and use only the copy. Since we have discovered that there is no actual loading of code during game play, we will "grab" the program code directly from the computer's memory and save it as a normal binary file. For this reason you should now format a normal DOS 3.3 48K slave disk with the hello program deleted. If you wish, you may use a fast DOS.

We will now need a method of resetting into the monitor. A brief discussion of this is found on the inside front cover of every COMPUTIST issue. I am going to describe an extra method, which involves the use of a common Language Card, or any other extra memory card (I personally use a 32K Saturn card).

The basic idea is to write-enable the card, copy the whole ROM onto it, modify the reset routine for jumping into the monitor and read-enable the card. Thus, when the Reset key is pressed during program execution, a monitor jump will be forced, since the computer thinks that the ROM is in the card.

Now the bad news. A lot of software protectors are aware of this trick. So they have included a routine in their programs to check the existence of a memory card in slot 0. If a card is found, it is read-disabled and the whole idea fails.

However, there is a way around this. We will move our card to another slot. If the check routine does not bother to examine slots other than 0, we will succeed. This is exactly the case with Cubit.

More bad news: Some commercial programs include routines that check the existence of memory cards in all slots. For example, Zenith, Horizon V, and Lady Tut do this. The following procedure is not applicable to these programs. If you have another way into the monitor, jump to step 7.

The Procedure

1) Turn off the power and move your RAM card to another slot. Slot 1 is what most people use, but I usually select slot 5 (the only one empty in my Apple).

2) Turn your computer on without booting. Press RESET to stop the drive and enter the monitor with **CALL -151**.

3) Two successive PEEKs to \$C081 will write-enable your language card in slot 0, so if you moved your RAM card to slot 1, you would type this from the monitor:

C091 N C091

This enables the Read ROM/Write RAM mode of your card. If you have selected a slot other than 1, you should consult the following table for the addresses you need.

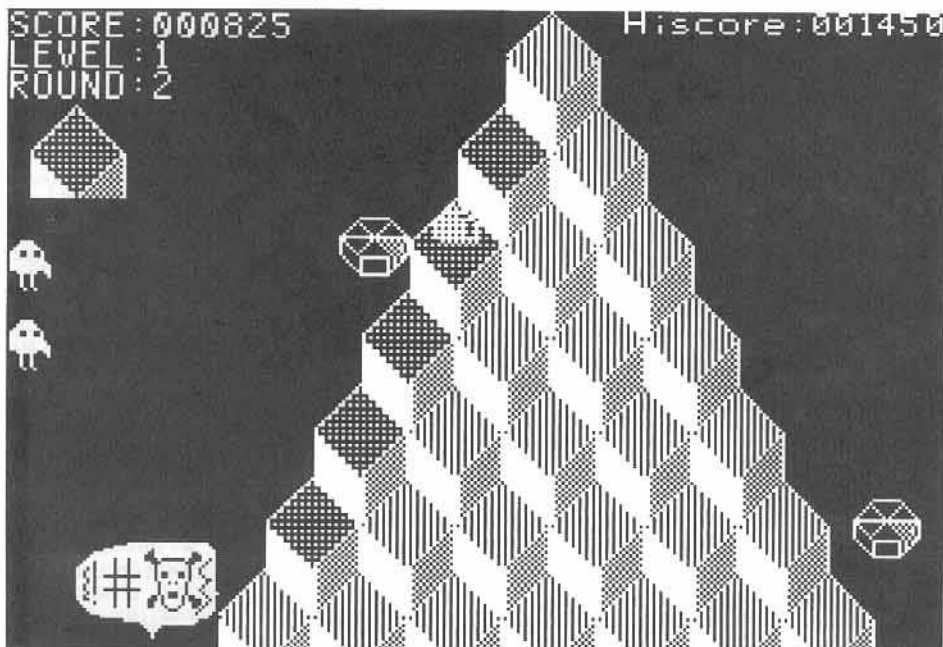
Card in Slot:	Read RAM:	Write RAM:
0	C080	C081
1	C090	C091
2	C0A0	C0A1
3	C0B0	C0B1
4	C0C0	C0C1
5	C0D0	C0D1

4) Now we will copy the image of the whole ROM to the card by "moving" it to the same place we read it from. When the move routine writes each byte to its "new" location, the RAM will accept it because we have allowed it to read now. Notice that the source and destination addresses are the same for this move command.

D000<D000.FFFF

5) This modifies the RESET routine in the F8 ROM for a direct jump to the monitor.

FA62: 4C 59 FF



6) A PEEK to \$C080 will disable the ROMs, read-enable the RAM, and write-protect the RAM on a slot 0 RAM card (yes, all at once). For slot 1, just look at \$C090 with the following command.

C090

Again, consult the table for another slot.

7) Insert the Cubit copy in your drive and boot it.

C600G

8) As soon as the game starts, press Reset to enter the monitor.

9) As mentioned previously, the RESET vector at \$3F2 and \$3F3 holds the program start address. You should find \$4000 there. However, the program code starts at \$19FD, where a JMP \$4000 occurs. The hi-res page 1

(locations \$2000 to \$4000) is the action screen. It is assumed that the code ends at \$9600, where the DOS buffers start.

10) The original disk check routine starts at \$9434. You can suspect that from the access at the RWTS parameter list (\$B7E8-\$B7F8) that follows. You must disable this routine by placing an RTS (\$60) at its start.

9434:60

11) In order to defeat the game restart when pressing Reset, we must modify the Reset fix routine, which lies at \$4069-\$4075.

406A:03

406F:E0

This will return you to Applesoft whenever Reset is pressed.

12) In order to save the (new improved) game, you must first boot your 48K slave disk. Insert the slave and type

C600G

You can now save Cubit as one binary file by typing:

BSAVE CUBIT,A,\$19FD,I,\$7C03

However, since this method involves the unnecessary saving of the action screen, you can try the following alternative:

BSAVE CUBIT1,A,\$1A00,L,\$600

BSAVE CUBIT2,A,\$4000,L,\$600

In this case, you must also write the following Applesoft program:

10 PRINT CHR\$(4)"BLOAD CUBIT1"

20 PRINT CHR\$(4)"BRUN CUBIT2"

Save this file as CUBIT LOADER (or HELLO) and RUN it in order to load Cubit. This method saves you 29 disk sectors.

13) You are now done. One final APT: the memory location \$4097 holds the number of Cubits (3 on the original). You can change this to a higher number.

P

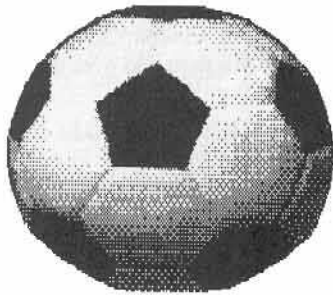
R

I

N

C

E



SOCCER

softkey for...

Baudville

by William Forsyth

*Baudville
1001 Medical Park Dr. SE
Grand Rapids, MI 49506*

Requirements:

64K Apple][Plus, //e, //c
DOS 3.3 or compatible
Super IOB 1.5 or a sector copier
A sector editor
Blank disks
Pixit, Take 1, or Prince

Baudville Software is among the best for creating graphic images, animating them, and printing them out in color to your printer. In this article, you will learn how to deprotect three of their most popular programs: Pixit, Take 1, and Prince. Furthermore, the softkey will be done in order of age of the programs, so that you can see how Baudville's protection schemes have evolved.

Pixit

Pixit is one of the oldest Baudville programs. This program allows you to create and manipulate shapes, which can be used in Take 1 or your own programs. It includes an excellent graphic editor with a wide assortment of various "pre-made" shapes, which can be further supplemented by purchasing their shape library disk or designing your own. If you boot up Pixit you will hear your drive arm make a series of rapid clicking noises, as if it were reading the tracks of the disk at a very high rate of speed. It is actually checking to see that all 35 tracks on the disk are perfectly in sync with track zero. You could get a working copy by using one of the popular bit copiers and copying tracks \$0-\$22 using the "synchronize tracks" option.

Fortunately, Baudville has left a big hole in its protection scheme, enabling the disk to be cracked rather easily. If you CATALOG the original disk, you will notice that everything is in FIDable form. These files do not check to see if the Baudville DOS is resident, therefore you can simply throw away the Baudville DOS and substitute your own!

Here's how:

- 1) First make a copy of your original Pixit by using COPYA or any other whole disk copier.
- 2) Boot up a normal DOS 3.3 disk. Remove this disk and put a blank in the drive.
- 3) Clear the memory by typing NEW.
- 4) Initialize the blank disk with the file name "MENU".

INIT MENU

- 5) Use the following Super IOB 1.5 controller to copy tracks 0-2 of the blank disk that you just initialized to the copy you made of Pixit. Do not format the copy.

```
1000 REM PIXIT CONTROLLER
1010 TK=0 : LT=3 : ST=15 : LS=15 : CD=WR : FAST
      =1
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 HOME : PRINT "DONE" : END
```

- 6) Relax and enjoy. Your copy should boot up and run cleanly. If you prefer, you could add a fast DOS like Pronto-DOS to make the boot even faster.

Take 1: (take two)

Take 1, like Pixit, uses synchronized tracks as its means of copy protection. However, on Take 1, only tracks 0 through 6 are synchronized. The program does not check the rest of the disk.

In COMPUTIST No. 25, a softkey for Take 1 was given. Unfortunately, the procedure will

P

I

X

I

T

Software

not work on all versions of this animation package. Another sector edit on track 0 is necessary to produce a working copy. Further exploration of the machine code revealed another routine that needs to be disabled.

This is accomplished by substituting two \$EA's for the two bytes in a conditional branch instruction at track 0, sector 6, byte \$2D. Thus, to get a completely unprotected Take 1, follow these steps.

- 1) Make a copy of the original Take 1 disk with COPYA or other normal disk copier.
- 2) Using a sector editor, make the following changes to the copy:

On track 0, sector \$6:

Byte#	From:	To:
\$16	\$D0	\$EA
\$17	\$F7	\$EA
\$2D	\$D0	\$EA
\$2E	\$D9	\$EA

On track 0, sector \$F:

Byte#	From:	To:
\$08	\$D0	\$EA
\$09	\$F7	\$EA

- 3) You are now the proud possessor of an unprotected copy of Take 1!

Prince

Prince is one of the newest releases from Baudville. This utility program allows you to transfer your hi-res and double hi-res pictures to your printer in color. This is accomplished by switching different colored ribbons in the printer during the dumping process. Also, if you use heat-transferable ink on your printouts, you can transfer your color screendump to a T-shirt! Prince also lets you make greeting cards and labels in color.

As for its protection, Prince uses synchronized tracks in a manner similar to Take 1. The tracks that are synchronized are 0 through 7 with one minor change, on Prince, if you take a look at the length of the tracks on

the disk (many bit copiers display track lengths) you will notice that they are much shorter than usual. This will prevent you from creating a bit copy backup with almost all newer bit copiers such as Locksmith 5.0 and Copy][Plus 5.0. However, the folks at Baudville, in designing their copy protection, did not consider older bit copiers such as Locksmith 4.1 that, in certain instances, tend to lengthen the tracks on the duplicate copy. As a consequence Locksmith 4.1, when told to synchronize tracks 0-7, will produce a completely working backup with the harmless side effect of having a more normal track length on the duplicate. However, Locksmith 5.0 and Copy][Plus 5.0 (the slick new versions) won't even get past the first track! The lesson here is not to throw out those old bit copiers just because you have the newer versions of them!

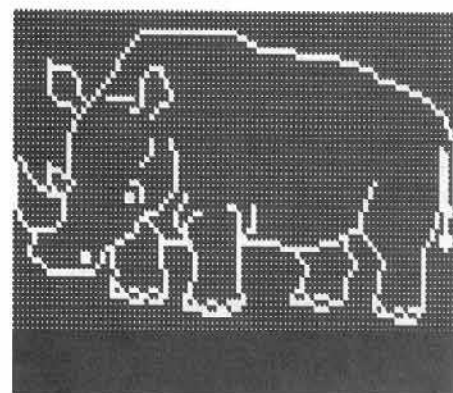
To disable the protection routines of Prince, we need not look very far, because they have used a RWTS very similar to that of Take 1 (it lives in the language card). The disk protection routine can be found on track 0, sector 6. The actual disk access is done by subroutines that this protection calls, so it is harder to locate as it contains nothing like LDA \$C08C,X to point directly to it. To disable this one, we will NOP out the Branch if Not Equal instructions that cause it to keep trying until a test is successful.

- 1) Make a copy of your original Prince disk with COPYA or similar.
- 2) Pull out your sector editor and make the following changes to your new copy.

Change track 0, sector 6:

Byte#	From:	To:
\$14	\$D0	\$EA
\$15	\$F7	\$EA
\$31	\$D0	\$EA
\$32	\$D5	\$EA

- 3) That's it. You're done. Note that the flip side, which contains the demo pictures, is not protected and can be copied normally.



Super IOB

by Ray Darrah

Requirements:

Apple II Plus, IIe or IIc
One DOS 3.3 disk drive

Throughout COMPUTIST, you will find several "Super IOB controllers." These short programs are intended to be keyed into a program called Super IOB. That is, the Super IOB program must be modified by adding lines (usually lines 1000 - 9999) to it. The resulting modified Super IOB program can then deprotect a particular program.

Since the introduction of Super IOB, COMPUTIST has used this flexible program to deprotect (or partially deprotect) dozens of commercial programs with far ranging protection techniques. In addition, COMPUTIST has printed articles designed to aid the programmer in his use of Super IOB. These include "CSaver" (COMPUTIST No. 13), a machine language program which helps in the merging of the controller with Super IOB, "The Controller Writer" (COMPUTIST No. 16), an Applesoft program which asks the user about the disk he is to deprotect and then formulates a Super IOB controller which will deprotect (or at least partially deprotect) that program, and "A Tutorial For: Disk Inspection And The Use of Super IOB" appearing in COMPUTIST No. 17.

Presented here is the latest version of Super IOB (v1.5). All previous controllers printed in COMPUTIST will work with this version. The only significant difference between this version of Super IOB and the version preceding it is the addition of a machine

language routine that allows a disk to be copied up to 350% faster.

Keying in Super IOB

Use the procedure outlined on the inside front cover of this magazine to key in the Applesoft portion of Super IOB and save it with:

SAVE SUPER IOB

Next (using the procedure for machine language on the inside front cover) key in the hexdump and save it with:

BSAVE IOB.OBJ0,A\$300,LSA8

The Function of Super IOB

Super IOB deprotects disks by using a modified RWTS (the subroutine in DOS which is responsible for the reading and writing of disk sectors) for reading from the protected disk and then using a normal RWTS for writing to the deprotected disk. Because of this, Super IOB can only copy disks with sectors that somewhat resemble normal DOS sectors. This includes a large percentage of commercial software currently on the market.

Before a disk can be deprotected by Super IOB, the protection scheme must be determined. After the protection is discovered, a controller is created for that specific protection.

The following is a discussion of various protection techniques and how Super IOB can be used to foil them.

Altered Marks

On every disk there are certain marker bytes that tell DOS where each sector starts and ends. For example, a normal 16-sector disk has the bytes D5 AA 96 at the beginning of every sector. These marker-bytes are put on the disk

only once (during initialization) and remain on the disk as permanent "milestones." A common practice (especially on older releases) is to change these markers to some other value such as D5 BB 96. A disk with such a change would only be copyable by a bit copier.

There are four markers for every sector on a disk. They are: start of address, end of address, start of data and end of data. Four subroutines in Super IOB modify the RWTS to look for markers of your choosing. These subroutines allow a normal RWTS to read an otherwise unreadable disk.

Halftracks

The Apple disk drive is capable of moving its read/write head to at least 70 distinct positions. However, because its read/write head is so wide it cannot reliably write data to each of these tracks (i.e. positions). Therefore, normal DOS only writes data on the even positions and calls them the 35 tracks. The odd positions have contracted the name of half tracks.

Sometimes a software publisher will put data on these so called half tracks which would once again render a disk uncopyable except by a bit copier. A subroutine in Super IOB will move the disk head to any position on the disk without letting the RWTS know of the move. This subroutine can therefore be used to copy disks that have data on those tracks between tracks.

Modified RWTS

If you are unable to determine the protection scheme being employed on a particular disk there is a catch all method. This method uses the "Swap RWTS at \$1900 with \$B800" routine in Super IOB and the swap controller presented later in this article.

version 1.5

The swap controller instructs Super IOB to read the protected disk with a foreign RWTS and write to a normal formatted disk with the standard RWTS. This method will work if the protected program has an RWTS at \$B800 with standard entry and exit conditions.

Complete instructions for using this catch all method appear after the swap controller in this article.

The Comprising Subroutines

Super IOB is little more than a collection of Applesoft subroutines. These subroutines are called by the inserted subprogram (the Super IOB controller) in a particular sequence to allow deprotection of a specific disk. Although the concept of Super IOB is simple, only by understanding the function of the subroutines in Super IOB can an Applesoft programmer learn to create a controller for a specific disk. The following is an explanation of the subroutines in Super IOB. During this explanation, I recommend that you refer to the listing of Super IOB to see how each subroutine is accomplished.

Name: **START UP**
Line Number(s): **10 - 60**
Entry Conditions: Not Applicable

The first few lines merely identify the program; however, line sixty sets HIMEM and LOMEM so that they fit the memory usage requirements (see Memory Map following). It then goes to "CONFIGURATION TIME".

Name: **INITIAL IOB SETUP**

Line Number(s): **80**

Entry Conditions:

DV= drive to be accessed
VL= volume of disk to be accessed
SO= slot to be accessed

This subroutine is normally GOSUBed via "TOGGLE READ/WRITE". Its purpose is to reset the buffer page and set the drive, slot and volume number to the disk to be accessed next.

Name: **R/W SECTOR**

Line Number(s): **100 - 110**

Entry Conditions:

TK= Track to be accessed
ST= Sector to be accessed
CD= Command code for the RWTS

This subroutine is GOSUBed directly from the controller. It reads or writes (depending upon CD) at the specified track and sector.

Name: **MOVE S PHASES**

Line Number(s): **130 - 140**

Entry Conditions:

SO= Slot of drive to move
DV= Drive number of drive to move
PH= Phase number that the disk head is currently over
S= Number of phases to move

This routine moves the disk read head the number of phases specified by S (one phase equals one half-track) and is capable of moving in either direction up to 128 phases (or 64 tracks). Care should be taken that PH + S isn't greater than 255 or less than 0 or an error will occur.

Name: **ALTERED ENDING MARKS**

Line Number(s): **170**

Entry Conditions: Proper DATA pointers

This routine changes the address field and data field epilogue markers in the normal RWTS. The values to change these to should be contained in a DATA statement. Because normal DOS only checks the first two bytes of these markers, only four values are required. The address field is changed first and should appear first in the data statement.

Name: **ALTERED ADDRESS MARKS**

Line Number(s): **190**

Entry Conditions: Proper DATA pointers

This routine modifies the RWTS (via POKE) so that it looks for a different sequence of address field prologue marks. The decimal values of the marks to look for should be stored as the next DATA elements.

Name: **ALTERED DATA MARKS**

Line Number(s): **210**

Entry Conditions: Proper DATA pointers

Same as previous subroutine except for DATA field prologue marks.

Name: **NORMALIZER**

Line Number(s): **230 - 250**

Entry Conditions: None

This routine restores the values in the RWTS subroutine that are changed by any routine in Super IOB. This routine should be called just before writing in order to fix the RWTS so that it can access normal DOS disks.

Name: **IGNORE ADDRESS CHECKSUM**
Line Number(s): **270**
Entry Conditions: none

This routine modifies the RWTS subroutine so that it doesn't examine the checksum byte of the address field. This routine has been incorporated in many controllers.

Name: **ALTERED DATA CHECKSUM**
Line Number(s): **290**
Entry Conditions: Proper DATA pointers

This routine alters the starting checksum byte that the RWTS subroutine will use when reading a DATA field. The normal value for the RWTS is 0. The value to change the checksum to should be the next DATA element.

Name: **THE SECTOR EDITOR**
Line Number(s): **310 - 340**
Entry Conditions: Proper DATA pointers and Elements

T1 = lowest track in buffer
TK = highest track in buffer

This routine automatically performs sector edits as the copy process goes on. It must be called (via GOSUB) just after reading a range of tracks. To indicate how many sector edits are to be performed, you must have a DATA element that has the number of sector edits followed by the word "CHANGES." For example:

```
1100 DATA 7 CHANGES,1,1,3,4
```

would tell the sector editor that the next 28 DATA elements are sector edits. This is because each sector edit is defined in four DATA elements. The location of the "x CHANGES" element in the DATA string does not matter because the sector editor will search it out and use the elements immediately following it.

The format for the four bytes that define a sector edit is: TRACK, SECTOR, BYTE, CHANGE TO. Each element is decimal and should be within the correct ranges since no error checking is done.

If you use the "R/W A Range Quickly" routine and you wish to perform some sector edits, you must set FAST equal to 1 so that this routine will be able to locate the specified sector in memory.

Name: **EXCHANGE RWTS's**
Line Number(s): **360**
Entry Conditions: A RWTS at \$1900

This is the standard swap RWTS's routine. It uses a routine in IOB.OBJ0 to exchange the RWTS at \$1900 with that which is located at \$B800, the normal location for an RWTS.

If you wish to exchange some other pages of memory, you may tell the machine language swap routine (invoked by a CALL 832) what to exchange by executing a few POKE's. They are as follows:

POKE 253, start of first location
POKE 255, start of second location
POKE 224, number of pages to exchange (a standard RWTS is eight pages long)

Name: **FORMAT DISK**
Line Number(s): **380 - 410**
Entry Conditions:
S2= slot of disk to format
D2= drive number of disk to format

This routine formats the target disk. It was meant to be used before the controller takes hold of Super IOB (and is GOSUBed by "Configuration Time") but can be called by the controller should the need arise.

Name: **PRINT TRACK & SECTOR #**
Line Number(s): **430**
Entry Conditions:

TK= The track number to display
ST The sector number to display

This is the subroutine that puts the current track and sector number at the top of the screen in hexadecimal during the softkey operation. It should be invoked just before reading or writing each sector.

Name: **CENTER MESSAGE**
Line Number(s): **450**
Entry Conditions: AS= The message

This routine prints a message in the center of the screen at the current VTAB position. Care should be taken that the message to print is not longer than 40 characters. If so, an error will occur.

Name: **PRINT MESSAGE AND WAIT**
Line Number(s): **470**
Entry Conditions: AS= The message

This routine uses "Center Message" to print the intended message at a VTAB of 11 and then it prints "PRESS ANY KEY TO CONTINUE." It waits for a keypress before RETURNing.

Name: **TOGGLE READ/WRITE**
Line Number(s): **490 - 530**
Entry Conditions: CD= current command code

This routine toggles the state of CD (from Read to Write and vice versa) and prints the current mode in flashing letters at the very top of the screen. In addition, if the user has only one drive, it asks him to swap disks. It then exits via "INITIAL IOB SETUP," thus making the sector buffer ready for the next operation.

Name: **IGNORE UNREADABLE SECTORS**
Line Number(s): **550 - 590**
Entry Conditions: Not Applicable

The controller should pay no attention to unreadable sectors, then somewhere in the beginning of it should be an "ONERR GOTO 550." This is used usually with RWTS.13 (since DOS 3.2 sectors are unreadable until they have been written to) but can be used with any

disk that has unreadable sectors which should be ignored.

Note that this routine will not function correctly if you are using "R/W A Range Quickly". To ignore errors when using this routine, insert a POKE 775,96 into the beginning of your controller.

Name: **R/W A RANGE**
Line Number(s): **610 - 620**
Entry Conditions:

TK,ST= first sector to read or write
I,T,LS last sector (minus one MOD 16) to read or write
MB= maximum buffer page, CD= command code

This routine quickly reads or writes a range of sectors by calling a machine language program. To be faster, this machine language program stores the sectors in memory in a decreasing manner. That is: sector \$0F is followed by sector \$0E and sector \$0D and so on.

Storing the sectors in memory in a different sequence than ascending is O.K. as long as, 1) the sectors are written in the same order they are read, and 2) "The Sector Editor" routine knows that the sectors were stored in descending order by setting the variable FAST equal to 1.

The Remainder Of The Program

Lines 1000 through 9999 are meant for the controller and all DATA statements it contains. All lines greater than 9999 are used by the error trapper or the configurer which consists of all the prompts when the program is run. The error trapper will print a disk error and stop the program. If the error wasn't a disk error, the error trapper will let it occur.

Now that you have an idea of the subroutines, take a look at the **Variable Usage** list on page 21 and note how the variables relate to them.

Memory Usage

Following is a memory allocation table for the various parts of Super IOB. It is extremely important to stay within the boundaries when writing a controller. If not, horrible things may happen (the least of which would be the production of an incorrect copy).

Memory Allocation Table

\$0800.\$18FF	(2048-6399)
intended for the Applesoft part of Super IOB	
\$1900.\$20FF	(6400-8447)
space allocated for a moved RWTS.	
\$2100.\$26FF	(8448-9983)
Super IOB Applesoft variable space	
\$2700.\$96FF	(9984-38655)
enough space for 7 tracks, this is the sector buffer	

First, notice the amount of space available for the BASIC program. This leaves ample room for most any controller.

Second, observe the 1534 bytes for variables. This should be enough space for the simple softkey procedure. It is impossible to allocate more memory for variables and use an alternate RWTS file. If you find that you need more memory and the program does not use RWTS.13 or some other moved RWTS, the **LOMEM: 8448** command may be removed from line 60. This will allocate what isn't used (by the BASIC program) of the 2K area reserved for the relocated RWTS as variable space.

Never omit the "HIMEM:" statement!

This could cause variables to overflow into the sector buffer, thus making a faulty copy.

With all this new knowledge, we are finally ready to scrutinize some sample controller programs. Keep in mind that protection schemes can be used with one another. Therefore, a more sophisticated controller for Super IOB will probably be required for most softkeys. Even so, developing new controllers isn't difficult.

The Standard Controller

LISTED below is the standard controller. This controller (when merged into Super IOB v1.5) can *only* copy normal DOS 3.3 disks. It is essential that you key this controller in and save it, as most controllers appearing in COMPUTIST are only modified versions of this controller.

fast controller

```
1000 REM FAST CONTROLLER
1010 TK=0:LT=35:ST=15:LS=15:CD=WR:FAST
    =1
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 HOME:PRINT"COPYDONE":END
```

Line Explanation

1000 - identifies the controller
1010 - initializes variables: TK = 0, ST = 15 - sets the starting sector to be copied at track 0, sector 15, LT = 35, LS = 15 - sets the last sector to be copied at track 34, sector 0, CD = WR - sets the command code to to write, FAST = 1 - tells the sector editor subroutine that the sectors are stored in memory in a decreasing order.
1020 - the read routine: begin by toggling read/write to read, read a chunk of sectors
1030 - the write routine: begin by toggling read/write to write, write a chunk of sectors, if last track was written, then go to exit routine.
1040 - update track and sector number for next read and go to read routine
1050 - the exit routine: clear the screen, print ending message and exit to Applesoft

The Swap Controller

This controller reads with one RWTS and writes with the normal RWTS. This controller

```
*-----*
*                                   *
*                                   Super IOB machine routines
*                                   *
*                                   BY RAY DARRAH
*                                   *
*-----*

03D9- RWTS.B800 .EQ $03D9 ENTRY POINT TO RWTS @$B800
D412- INVOKERROR .EQ $D412 ROUTINE THAT CAUSES BASIC TO DO THE ERROR CONTAINED IN X
1E00- RWTS.1900 .EQ $1E00 ENTRY POINT TO THE RWTS AT $1900
B9A0- SEEKABS .EQ $B9A0 ENTRY POINT TO THE SEEKABS ROUTINE AT $B800
DE- BAS.ERR .EQ 222 BASIC ON ERR ERROR CODE
FC- SWFRM .EQ $FC EXCHANGE FROM PARAMTER
FE- SWTO .EQ $FE EXCHANGE RWTS 'TO' PARAMETER
E0- PAGES .EQ $E0 NUMBER OF PAGES OF MEMORY TO EXCHANGE
24- CH .EQ $24 CURSOR X POSITION
FDDA- PRNTBYTE .EQ $FDDA PRINTS HEXADECIMAL BYTE
```

```
.OR $0300 STARTS AT PAGE THREE
.TF 10B.0BJ0
```

```
*-----*
*                                   *
*                                   CALL RWTS
*                                   *
*-----*
```

```
0300: A9 03 IO LDA #TABLETYP ENTRY POINT TO THE RWTS THROUGH BASIC
0302: A0 0A LDY #TABLETYP A,Y POINT TO THE IOB TABLE
0304: 20 D9 03 JSR RWTS.B800 GO TO THE RWTS AT $B800
0307: B0 16 BCS DOS.ERR IF THE CARRY SET THEN CAUSE BASIC ERROR
0309: 60 RTS OTHERWISE, ALL IS WELL SO RETURN
030A: 01 TABLETYP .HS 01 TYPE OF TABLE (1=IOB)
030B: 60 SLT .HS 60 SLOT TO BE ACCESSED NEXT (VIA POKESLT.S0)
030C: 01 DRV .HS 01 DRIVE TO BE ACCESSED NEXT (1 OR 2)
030D: 00 VOL .HS 00 VLUME TO BE ACCESSED (0=ANYTHING WILL DO)
030E: 00 TRK .HS 00 TRACK TO ACCESS
030F: 00 SCT .HS 00 SECTOR TO ACCESS
0310: 1B 03 DCTPTR .DA DCT POINTER TO THE DEVICE CHARACTERISTICS TABLE
0312: 00 BUFFERLO .HS 00 ALWAYS MAKE LSB OF BUFFER POINTER ZERO!
0313: 27 BUF .HS 27 SECTOR BUFFER PAGE POINTER
0314: 00 NOTHING .HS 00 NOT USED
0315: 00 BYTCOUNT .HS 00 BYTE COUNT FOR PARTIAL SECTOR (0=256 BYTES)
0316: 00 CMD .HS 00 COMMAND CODE (0=SEEK)
0317: 00 RWTS.ERR .HS 00 ERROR CODE THA THE RWTS.B800 RETURNS WITH
0318: 00 OVL .HS 00 VOLUME NUMBER OF LAST ACCESSED DISK
0319: 60 OLDSLT .HS 60 SLOT PREVIOUSLY ACCESSED
031A: 01 OLDDRV .HS 01 DRIVE PREVIOUSLY ACCESSED
031B: 00 DCT .HS 00 DEVICE TYPE OF DEVICE CHARACTERISTICS TABLE
031C: 01 PHASES .HS 01 PHASES-1 PER TRACK. (0 OR 1)
031D: EF D8 MOTORCNT .HS EFD8 MOTOR-ON TIME COUNT
031F: AD 17 03 DOS.ERR LDA RWTS.ERR DOS HAS HAD AN ERROR. GET THE ERROR CODE
0322: 4A LSR DIVIDE IT BY 16
0323: 4A LSR
0324: 4A LSR
0325: 4A LSR
0326: AA TAX TRANSFER IT TO X SO BASIC WILL ERROR
0327: 4C 12 D4 JMP INVOKERROR CAUSE A BASIC ERROR
```

```
*-----*
*                                   *
*                                   MOVE THE DISK ARM
*                                   *
*-----*
```

```
032A: A9 00 MOVPHASES LDA #000 SET UP THE REGISTERS BEFORE CALLING SEEKABS
032C: A2 00 LDX #000 X AND A HAVE DUMMY NUMBERS THAT WILL BE POKE
                                          D INTO BY "MOVE S PHASES"
032E: 4C A0 B9 JMP SEEKABS
```

```
*-----*
*                                   *
*                                   CAUSE ERROR IN CONTROLLER
*                                   *
*-----*
```

```
0331: A6 DE BASICERR LDX BAS.ERR CAUSE THE ERROR # AT 222
0333: 4C 12 D4 JMP INVOKERROR
```

is very handy and used by Hardcore COMPUTIST frequently.

swap controller

```
1000 REM NEW SWAP CONTROLLER
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 "COPYDONE":END
10010 PRINT CHR$ (4) "BLOAD^RWTS,XXX,AS1900"
```

To use this controller, you must first boot up your protected disk, pop into the monitor and move its RWTS into a safe area of memory by typing:

1900<B800.BFFFM

Next, you must boot up a normal DOS 3.3 disk (preferably your Super IOB disk) and save this relocated RWTS

BSAVE RWTS,XXX,AS1900,LS800

It is recommended that you change the filename from "XXX" to that of the program you are trying to deprotect. You will then have to change the "XXX" in line 10010 of the swap controller to whatever you change the "XXX" to in the above command.

The last step in using the swap controller is to merge it with the Super IOB v1.5 body and RUN it.

Saving the Controller

It is recommended that you have copies of both of these controllers. The best method of controller manipulation (typing in saving, merging with Super IOB etc.) is through the use of "CSaver" which appeared in COMPUTIST No. 13.

Now go out there and break some disks!



Super IOB Hexdump

```
0300: A9 03 A0 0A 20 D9 03 B0 $BD35
0308: 16 60 01 60 01 00 00 00 $9CF5
0310: 1B 03 00 27 00 00 00 00 $4320
0318: 00 60 01 00 01 EF D8 AD $55A7
0320: 17 03 4A 4A 4A 4A AA 4C $B42B
0328: 12 D4 A9 00 A2 00 4C A0 $8038
0330: B9 A6 DE 4C 12 D4 68 A8 $6E1C
0338: 68 A6 DF 9A 48 98 48 60 $FDD9
0340: A0 00 84 FC 84 FE B1 FC $3777
0348: 48 B1 FE 91 FC 68 91 FE $AAB9
0350: C8 D0 F3 E6 FD E6 FF C6 $921F
0358: E0 D0 EB 60 97 35 0F A9 $5F1E
0360: 00 85 28 A9 05 85 29 A9 $8FFA
0368: 10 85 24 AD 0E 03 20 DA $41EA
0370: FD A9 1C 85 24 AD 0F 03 $B401
0378: 20 DA FD 20 00 03 CE 0F $011B
0380: 03 10 08 A9 0F 8D 0F 03 $3519
0388: EE 0E 03 EE 13 03 AD 13 $387C
0390: 03 CD 5C 03 B0 C5 AD 0F $EFC A
0398: 03 CD 5E 03 D0 C9 AD 0E $18A0
03A0: 03 CD 5D 03 90 C1 B0 B3 $250A
```

```
*-----*
*                POP OFF RETURN                *
*-----*
0336: 68      POP      PLA ROUTINE TO POP OFF ONE RETURN (BASIC) ADDRESS
0337: A8      TAY
0338: 68      PLA
0339: A6 DF    LDX BAS.ERR+1 FIX STACK
033B: 9A      TXS      PUT THAT AS THE STACK POINTER
033C: 48      PHA
033D: 98      TYA      RESTORE THE LAST RETURN ADDRESS.
033E: 48      PHA
033F: 60      RTS

*-----*
*                EXCHANGE RWTS's               *
*-----*
0340: A0 00    LDY #0      ZERO THE LSB's
0342: 84 FC    STY SWFRM  AND HAVE Y AT ZERO FOR START
0344: 84 FE    STY SWTO
0346: B1 FC    MOVE PAGE LDA (SWFRM).Y GET A BYTE
0348: 48      PHA      AND SAVE IT
0349: B1 FE    LDA (SWTO).Y GET THE BYTE WHERE THE SAVED ONE GOES
034B: 91 FC    STA (SWFRM).Y AND STORE IT WHERE THE SAVED ONE WAS
034D: 68      PLA      GET THE SAVED BYTE
034E: 91 FE    STA (SWTO).Y AND STORE IT WHERE IT GOES
0350: C8      INY      DONE WITH A PAGE
0351: D0 F3    BNE MOVE.PAGE NO KEEP WORKING ON IT
0353: E6 FD    INC SWFRM+1 GET NEXT MSB's
0355: E6 FF    INC SWTO+1
0357: C6 E0    DEC PAGES  DECREMENT THE NUMBER OF PAGES TO MOVE
0359: D0 EB    BNE MOVE.PAGE IF NOT DONE, MOVE ANOTHER PAGE
035B: 60      RTS1     RTS      FINISHED, RTS

*-----*
*                READ OR WRITE THE ENTIRE BUFFER *
*-----*
035C: 97      MB      HS 97      HIGHEST BUFFER PAGE+1
035D: 35      LT      HS 35      LAST TRACK TO GET+1
035E: 0F      LS      HS 0F      (LAST SECTOR TO GET-1)MOD16

035F: A9 00    LDA #0      DO A VTAB3
0361: 85 28    STA $28
0363: A9 05    LDA #5
0365: 85 29    STA $29
0367: A9 10    NEWTRK LDA #16     HTAB
0369: 85 24    STA CH
036B: AD 0E 03 LDA TRK
036E: 20 DA FD JSR PRNTBYTE PRINT IT
0371: A9 1C    LDA #28     HTAB
0373: 85 24    STA CH
0375: AD 0F 03 LDA SCT
0378: 20 DA FD JSR PRNTBYTE
037B: 20 00 03 JSR IO      GET A SECTOR
037E: CE 0F 03 DEC SCT     NEXT SECTOR.
0381: 10 08    BPL NXT.PG
0383: A9 0F    LDA #15     RESTORE TO SECTOR 15
0385: 8D 0F 03 STA SCT
0388: EE 0E 03 INC TRK     NEXT TRK
038B: EE 13 03 NXT.PG INC BUF     NEXT PAGE OF MEMORY
038E: AD 13 03 LDA BUF
0391: CD 5C 03 CMP MB
0394: B0 C5    BCS RTS1    BUFFER FULL. RETURN
0396: AD 0F 03 LDA SCT     LAST SECTOR?
0399: CD 5E 03 CMP LS
039C: D0 C9    BNE NEWTRK
039E: AD 0E 03 LDA TRK
03A1: CD 5D 03 CMP LT
03A4: 90 C1    BCC NEWTRK
03A6: B0 B3    BCS RTS1    END OF DISK
```

Super IOB Variables

A—General temporary usage, scrambled by "Move S Phases" and "The Sector Editor."

AS—holds message to pass to the user via "Center Message" and "Print Message And Wait" and is scrambled by "Toggle Read/Write."

A1, A2, A3, A4—scrambled by "Altered Address Marks", "Altered Data Marks", "The Sector Editor", "Altered Ending Marks" and "Altered Data Checksum" these are READ from DATA statements and POKEd into the appropriate RWTS to change it.

BS altered only by "Configuration Time."

BF—Buffer Full holds the status of the sector buffer and is set to 1 if the buffer is either full or empty and 0 if neither. Is changed only by "R/W SECTOR."

BUF—BUFFer constant holds the address where the RWTS is expecting to find the page number of the sector and is used by "Initial IOB Setup" and "R/W Sector". A PEEK(BUF) will return the current sector buffer page number.

CD—CommanD code is used by the controller, "Toggle Read/Write" and "R/W Sector" and holds the current RWTS command code (see RD, WR, and INIT).

CMD—CoMmanD code constant holds the address where the RWTS is expecting to find the previously stated command code and is used by "R/W Sector". A POKE CMD,CD will change the IOB command.

D1—Drive 1, set during configuration to the drive number of the source drive, is used by "Toggle Read/Write."

D2—Drive 2, same as above except for target drive.

DOS—Disk Operating System specifies the number of sectors to read or write and is initialized to 16.

DRV—DRiVe constant holds the address where the RWTS is expecting to find the drive number of the drive to be accessed and is used by "Initial IOB Setup". A PEEK(DRV) will return the drive last accessed.

DV—current DRiVe, used by "Initial IOB Setup", "Toggle Read/Write", and "Move S Phases". holds the drive number of the drive to be accessed next.

ERR—ERRor code is used by "Disk Error" to determine the error that has just occurred.

FAST—used by "The Sector Editor" to calculate the correct addresses of specified sectors if set to 0. The "The Sector Editor" then assumes that all sectors are stored in consecutive memory pages in an ascending order. If set to a 1, "The Sector Editor" routine assumes that the sectors are stored in a decreasing order (see the section describing the operation of "R/W A Range of Sectors").

INIT—INITIALize command code. A CD=INIT will set the command code to format the diskette.

IO—Input/Output constant holds a 768 (set during configuration) and is CALLED by "R/W Sector" to induce the RWTS subroutine.

LS—Last Sector number is used to tell "R/W A Range Quickly" what sector is the last sector to be read.

LT—Last Track number is used to tell "R/W A Range Quickly" the last track to be read or written.

MB—Maximum Buffer page holds the last page of memory for the sector buffer, is used by "R/W Sector", is initialized (during configuration) to 151 and should be changed to 130 only when a 13-sector disk is read or written.

OVL—Old VoLume constant. A PEEK(OVL) will return the volume number of the previously accessed diskette.

PH—current PHase. If "MOVE S PHASES" is referenced (by the controller), this variable must contain the disk arms current phase number ($PH=2*TK$).

RD—ReAD command code. A CD=RD will set the command to read the disk.

S—Step is used to tell "Move S Phases" how many phases to step through (-120 to 120).

S1—Slot 1 means to set to the slot number of the source drive during configuration and is used by "Toggle Read/Write".

S2—Slot 2. Same as above except for target drive.

SCT—SeCTOR number constant holds the address where the RWTS is expecting to find the sector to be accessed and is used by "R/W S" to tell the RWTS which sector is to be read or written. A PEEK(SCT) will return the last accessed sector number.

SLT—SLoT number constant holds the address where the RWTS is expecting to find the slot number of the disk to be accessed next and is used by "Initial IOB Setup". A PEEK(SLT) will return the last accessed disk slot number.

SO—SIoT number is used by "Toggle Read/Write" and "Initial IOB Setup" and holds the slot number of the disk to be accessed next.

ST—SecTOR number is used by the controller to tell "R/W Sector" which sector number is to be read or written next and is also used to tell "R/W A Range Quickly" the starting sector to be read or written.

TK—TRacK number is used by the controller to tell "R/W Sector" which track is to be accessed next and is also used to tell "R/W A Range Quickly" the starting track to read or write.

TRK—TRacK number constant holds the memory location where the RWTS is expecting to find the track to be accessed. A PEEK(TRK) will return the last accessed track number.

VL—VoLume number is used by the controller to tell "Toggle Read/Write" (which passes it to "Initial IOB Setup") the volume number of the disk to be accessed next.

VL\$—altered only by "Format Disk".

VOL—VOLume number constant holds the memory location where the RWTS is expecting to find the volume to be accessed. A PEEK(VOL) will return the volume number last used by the controller.

WR—WRite command code. A CD=WR will set the command to write.

Hartley Software

by Dave Stanton

Hartley Courseware, Inc.
(517) 646-6458

Requirements:

48K Apple II or equivalent

Blank disks

Super IOB 1.5

A way into the monitor, or

128K //e or //c with XFER.BOOT/RESTORE

Original Hartley disks

Few educational software companies have been more prolific than Hartley. Though many of Hartley's programs were written in the early eighties, they continue to be staples in the schools.

The programs generally lack fancy graphics and are quite slow in operation. However, they are easy to run, and they provide for teacher-modification. As drill and practice tools, they offer many advantages over the traditional pen and paper process.

An especially noteworthy feature of the series is its student record keeping. The program maintains a record of results for each person who tries it. Those results can be accessed by the teacher or parent as necessary.

As with any purchased software, prudence requires that a backup be kept safely tucked away for emergencies. While most copiers will handle Hartley software, these softkeys will permit you to analyze and modify the programs and textfiles.

For de-protection purposes, the company's programs can be classified into two groups: (1) the "Medalist Series" and other recent programs; (2) "Verbs," "Synonyms and Antonyms," and other pre-1984 programs.

Hartley uses two distinct protection schemes on its releases. Neither is very complicated, and the one used on the "Medalist Series" is the simplest.

The Medalist Series

"The Nibbler" (COMPUTIST No. 19) reveals essentially standard tracks and sectors with minor changes in the address field and data

field epilogues. The standard DE AA bytes have been changed to DA AA.

If you wish to check, you will notice unusually long sync fields between some sectors. Those fields often contain non-standard sync bytes (something other than FF). This should pose no problem.

Finally, some sectors appear to be normal at first, but they cause read errors when Super IOB tries to copy them. Although there could be many reasons for this, further analysis is unnecessary, since a perfectly functioning copy can be produced by activating the ignore unreadable sectors routine.

The Procedure:

1) Initialize a disk.

INIT HELLO

2) Install Hartley Controller A below into Super IOB 1.5 and RUN it.

3) Ignore the disk rattle. Some disks copy quietly; others protest.

This technique will also work on "Capitalization Practice" and "Capitalization Test." If you have other relatively recent releases, you may want to try this first.

controller A

```
1000 REM HARTLEY CONTROLLER A
1010 POKE 775,96:TK=3:LT=35:ST=15:LS=
15:CD=WR:FAST=1
1020 RESTORE:GOSUB 170:GOSUB 490:GOSUB 610
1030 GOSUB 230:GOSUB 490:GOSUB 610:IF PEEK
(TRK)=LT THEN 1050
1040 TK=PEEK(TRK):ST=PEEK(SCT):GOTO 1020
1050 HOME:PRINT "COPY^ DONE!":END
5000 DATA 218,170,218,170
```

Pre-1984 Hartley Software

The early programs from Hartley appear to be on slightly modified DOS 3.2 disks. Checking the raw nibbles with The Nibbler shows that each track contains only 13 sectors. Furthermore, the address field prologue is D5 AA B5, normal for DOS 3.2.

Since the address and data field epilogues vary from sector to sector, we will capture the RWTS and use a customized swap controller to load from the original disks.

Copying a 13 sector disk with Super IOB 1.5 requires two unique changes. The values of ST and LS must be changed to 12, and a modification of the machine language subroutine must be made. **POKE 900,12** tells

the ML routine to read only thirteen sectors instead of the usual sixteen.

The Procedure:

1) Initialize a blank disk with DOS 3.3.

INIT HELLO

2) Get your favorite way into the monitor ready, or load XFER.BOOT (COMPUTIST No. 28, pg. 15) and use it to boot the program into the extended 80 column card of your //e or //c.

```
BLOAD XFER.BOOT
PR#3
CALL 768
```

3) Stop the boot by opening your disk drive gate as soon as you see the first screen.

4) Hit Reset. Load and install RESTORE, XFER.BOOT's companion program.

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

5) Move the trapped RWTS into lower memory at \$1900.

```
1900<B800.BFFF
```

If you didn't use the RAM card method, type this instead.

```
1900<B800.BFFFM
```

6) Save the protected RWTS on your Super IOB 1.5 disk.

```
BSAVE HARTLEY.RWTS,$1900,$800
```

7) Run Super IOB with Controller B below installed.

controller B

```
1000 REM HARTLEY CONTROLLER B
1010 TK=3:LT=35:ST=12:LS=12:CD=WR:FAST
=1:POKE 900,12:POKE 775,96
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)"BLOADHARTLEY.RWTS.
A$1900"
```

In addition to the peace of mind that backups provide, you will now be able to modify your programs, put them on a hard disk, or do whatever you wish to improve them. Enjoy!



by Dick Meikle

Sun Microsystems, Inc.
P.O. Box 1388
Fort Lauderdale, FL
\$59.95

Requirements:

Apple][Plus or equivalent
A blank disk
FID or similar

Bridge is a utility that transfers PFS Series files to DOS 3.3 text, DIF, or Applewriter][mail list files. Those of us who now use Appleworks for most data base, word processing and spreadsheet work have found this utility an absolute necessity for transferring our old PFS: File records to ProDOS format. Bridge cannot transfer files directly from PFS (Pascal) files to ProDOS. First the PFS files must be converted to DOS 3.3 text or DIF Files with the Bridge. Then these files are converted

LOAD BRIDGE, A\$800

Then I entered the monitor.

CALL -151

After trying several locations by using the monitor G(0) command, I tried E00G. When I did this, the drive turned on and the head moved to track \$23 to start its search for the first three bytes at Sector \$0F. I then knew that I had found the entry point for the protection scheme. At this point, all I needed to do was disarm the Track \$23 search. When I listed the program at SE0B,

```
E0B- LDA $0800  Check $800
E0E- CMP #$4C   Is it a $4C? (a JMP)
E10- BNE $0E15  Then go to protection
E12- JMP $0EF9  Else start program
E15- LDA #$15   Set up Reset vector
E17- STA $03F2  (to point to $E15)
E1A- LDA #$0E
E1C- STA $03F3
E1F- EOR #$A5
E21- STA $03F4
E24- LDY #$23   Track & sector
E26- LDA #$0F   to read
E28- LDX #$01
```

E00G and the main screen appeared. I ran the program through a series of tests to see if it operated properly. The program appeared to work flawlessly. This was the entry point I was looking to find. There was only one more problem to eliminate. After making sure the three bytes SAD \$xx \$yy at \$800-\$802 were not necessary for program operation, I changed these bytes to Jump to \$0E00.

800:4C 00 0E

This is necessary because if you use the "change menu" option at any time, Bridge is rewritten to the disk with a BSAVE BRIDGE, A\$800, L\$2400. If the JMP \$0E00 is placed at \$7ED, 3 bytes before \$800, the JMP will not be written back to the disk. The next time Bridge is BRUN the Applesoft prompt will appear and Bridge will not work. Putting the JMP at \$800 eliminates this problem.

Steps for the Softkey

1) Boot DOS 3.3 and initialize a blank DOS 3.3 disk.

```
PR#6
NEW
INIT HELLO
```

softkey for BRIDGE

to ProDOS using the ProDOS users disk.

Bridge can be copied (EDD, Copy][, etc...) by copying tracks 0-\$23. Track \$23 is not usually copied by most copy programs. On track \$23, sector \$F, there are 3 bytes necessary for the program to boot. If byte \$02 on track \$23, sector \$F is greater than \$00 you are permitted to make a copy through the built-in copy program. On the original, this byte is a \$01. After you make one backup, this byte is changed to a \$00. Therefore, you will not be able to make another copy. Initially, I changed this byte to \$FF. This way I could make 255 copies from the original, as this byte will be decremented by \$01 each time a copy is made.

Like most people, I like to have many utilities on a single DOS 3.3 disk. Bridge is a total load program, so I set out to make a normal BRUN file. I tried to BRUN BRIDGE with DOS 3.3, but after loading, I received an Applesoft prompt and it ceased execution. This told me that Bridge does not start at \$800 (which is where it starts to load). Feeling that if I could find the program entry point, it might be able to run under DOS 3.3, I booted DOS 3.3 and loaded Bridge from my original.

```
E2A- JSR $0836  Gosub read routine
E2D- BCS $0E53  If read bad then crash
E2F- LDA $083F  Store the address of
E32- STA $06    the three bytes from
E34- LDA $0840  trk $23, sct $F in zero
E37- STA $07    page locations $6-$7
E39- LDY #$00   Set Y index to 0
E3B- LDA ($06),Y Get first byte
E3D- CMP $801   Compare to val at $801
E40- BNE $0E53  If no match then crash
E42- INY       Next Y
E43- LDA ($06),Y Get 2nd byte
E45- CMP $802   Compare with $802
E48- BNE $0E53  If no match, crash
E4A- INY       Next Y
E4B- LDA ($06),Y Get third byte
E4D- STA $29DD  Store it at $29DD
E50- JMP $0EEA  Start the program
```

I noticed that it starts by loading the accumulator with the byte at \$800 and comparing it with the number \$4C to see if it is a JMP. On my copy \$800 contained SAD, so this caused a Branch if Not Equal (BNE) at \$E10. I NOPed (No Operation) the BNE.

E10: EA EA

This way the branch is not taken to the protection routine starting at \$E15. I again tried

2) BLOAD Bridge from the original.

BLOAD BRIDGE, A\$800

3) Enter the monitor.

CALL -151

4) NOP the Branch to the protection scheme and set the beginning to the program entry point.

```
E10: EA EA
800: 4C 00 0E
```

5) Save Bridge back to the DOS 3.3 disk.

BSAVE BRIDGE, A\$800, L\$2400

6) Transfer the MAILLIST file from the Bridge disk to the DOS 3.3 disk using the file FID.

You now have a COPYable and Brunable Bridge. Bridge may now be transferred to any DOS 3.3 disk.

The file MAILLIST is a WPL (word processing language) file used only for Applewriter][mail list usage. If you don't use Applewriter][for merging a mailing list with a letter it will not be necessary to transfer MAILLIST to the slave disk.



Microzines

by Wesley R. Felty

In COMPUTIST No. 27, there are three articles on de-protecting Microzines. The first two softkeys using the Super IOB program didn't work at all for my versions of Microzines Volume 1, numbers 2-6. Eric Ondler's method (page 14) did work, but unfortunately, none of the methods given would allow the de-protected Microzines to create or use data disks. This article will give the general technique needed to patch the Microzines and specific fixes for Microzines 2-6 (Volume 1).

First of all, let's look at the problem. The original protected Microzines all run on a non-standard DOS. In order for them to use a data disk with normal DOS, each Microzine's special DOS includes special routines (located around 46748) to switch back and forth from the special DOS to a normal DOS. Therefore, each of the Microzine programs that needs data disks uses CALLs to activate one DOS or return to the other. The CALLs are to variable names like "DOS", "DS", and "DO" to turn on the data disk DOS and "XD" or "XDOS" to return to the Microzine's modified DOS. But when we de-protect the Microzines, we replace the special DOS with a normal one, so all of the DOS switching CALLs "bomb" and exit the program in the monitor.

To be able to use data disks with de-protected Microzines, we have to determine which programs in each Microzine use them, find and eliminate the CALLs to switch DOSes within them, and eliminate the DOS switching CALLs from the program on each Microzine disk that creates the data disks. The program descriptions usually tell us if a data disk can be used with each program, but it is a little trickier to find the actual program to patch since the Microzine may load three or four other programs before it gets to the real one. It usually loads graphics screens, machine language routines, instruction programs, etc. before it runs the real program. To find the actual programs that contain the code to be patched, boot the de-protected Microzine and when the main menu appears, press Reset. Then type "MONICO" (see your DOS manual) and then "RUN." Now, as each file or program gets loaded, a message will be printed to the screen showing the full DOS command loading or running it. Just run the programs normally until the program bombs into the monitor after asking you to insert the data disk. The last program that was RUN is

the one to patch. To turn off the MON mode, you can just hit Reset, or enter "NOMONICO" from immediate mode.


It is fairly easy to find the CALLs to look for if you have the proper tools. One of the first several lines will have code like "DOS = 46784: XDOS = DOS + 6", showing what parameters switch the DOS back and forth. If you have an Applesoft program line editor, like GPLE, it shouldn't be too hard to find the lines with the CALLs and delete them. Leave the rest of the line as it was and don't delete any calls other than the calls to switch DOS. In most lines, both CALLs are in the same line, usually surrounding commands to "OPEN" or "CLOSE" and "READ" or "WRITE" text files. If the second CALL isn't in the same line as the first one, then it should be somewhere in the next few lines. You may have to do a manual search of the longer lines if your editor has trouble with packed lines. If you leave some of the "CALL DOS" or "CALL XDOS" CALLs in, then the program will bomb into the monitor when it gets to them at a later time, so be sure to thoroughly test the data disk routines later. Finish up by saving the Applesoft program back to disk. Below are lists of the program lines that needed to be changed on my Microzines, including the names of the variables used to switch DOSes. On disk #2, the Melody Maker program is the only one needing a data disk, and it didn't use the calls so no patching is needed with it.

Using Data Disks That Need Changes

Vol 1, Disk 3:
program RR MAIN, variables DOS, XD
6020, 6030, 6040, 6050, 6060, 6230, 6240,
6250, 6280, 6410, 6420, 6430, 6450, 6620,
6630, 6905, 6910, 6920, 6930, 6940, 20000,
4415 Change to 4415 REM

Vol 1, Disk 4:
program LOG, variables DS, XD
46, 90, 96, 324, 330, 360, 378, 406,
416, 422, 432, 642, 648, 658, 672, 884,
13000

Vol 1, Disk 5:
program TR MAIN, variables DO, XD
6905, 6805 change to 6805 REM

Vol 1, Disk 5:
program MAKEABOOK, variables DO, XD
1110, 5020 (type a  before DELETE)
3960, 8100, 8125, 8205, 8225, 8310, 8400,
9318, 10984, 11000, 20000, 61215, 61230

Vol 1, Disk 6
program OLYMPIC RECORDS, vars DOS, XD
8010, 8030, 8050, 8070, 8085, 8090, 8100
61420, 61430, 61620, 61630

Vol 1, Disk 6
program SOUND OFF!, variables DOS, XDOS
5025, 8018, 8204, 8320, 9300, 9318,
10982, 10988

Finally, we need to change the program on each Microzine disk that creates the data disks. Actually, we could fix just one of them and use it to produce data disks for all of the other Microzines, but I hate to have partially working disks around, so I'll show the changes needed on Volume 1, Microzine numbers 2-6. Fortunately, on all of the Microzines, the data disks are produced in a program named "UTILITIES" by BLOADing a file called "INIT.OBJ" at \$9300 and issuing a "CALL 37632". All we have to do with each Microzine is to "LOAD UTILITIES", delete the CALLs to switch DOSes that surround the "CALL 37632", and "SAVE UTILITIES" back to the disk. In Microzine #2, no changes are needed at all. In Microzines #3 and #4, just delete lines 3017 and 3024 and "SAVE UTILITIES" back to disk. And in Microzines #5 and #6, just delete the "CALL DO" and the "CALL XD" from line 3010 and save UTILITIES back to the disk. That's it! We can now create and use data disks with all of the Volume 1 Microzines. I'm sure that the general method will also work with all of the other Microzines.

Recovering APEX

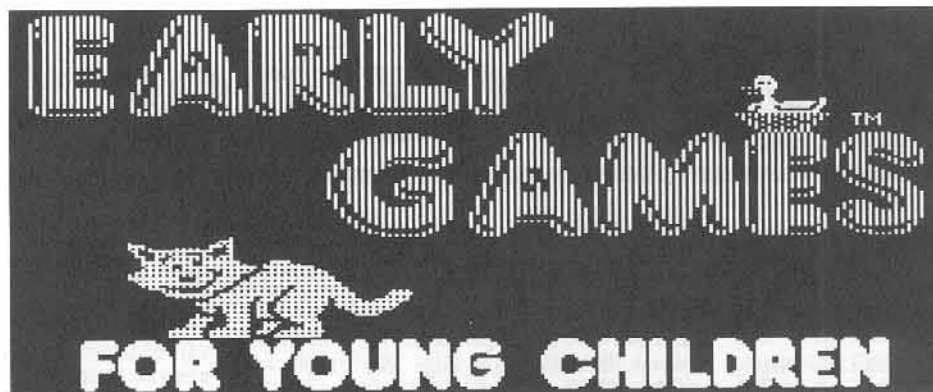
In COMPUTIST No. 27 (page 10), L.A. Carreira gave instructions to copy the file "APEX" or "HRCG" from a later Microzine back to Microzines #1 and #2. But what if you don't have a later Microzine? It is possible to recover the code from the original Microzine if you have a method of resetting out of the program into the monitor after the title page gets displayed. This takes a hardware modification such as the Don Lancaster absolute reset (which I highly recommend) or an old Integer Card. Then just put the de-protected disk in the drive and enter

BSAVE APEX, ASE00, L\$1200

And, finally, add a line like

48 PRINT CHR\$(4)"BLOAD APEX"

into the HELLO program.



by Joe Drelling

Springboard Software, Inc
7807 CreekrIDGE Circle,
Minneapolis, MN 55435
(612) 944-3912
\$39.95

Requirements:

Apple II Plus and up
A slave disk with no HELLO program
One blank disk
Super IOB 1.5
Early Games for Young Children

Early Games for Young Children (Early Games) is another popular educational game for small children. The game is supposed to be designed for children aged 2 1/2 to 6 but it keeps my one-year-old son constructively entertained (with some assistance). Early Games has nine educational games including: Number Matching, Counting, Addition, Subtraction, Letter Matching, the Alphabet, Names, Comparing Shapes, and Picture Drawing. Though the programs can be easily run on a monochrome monitor, all of the games take full advantage of the Apple's color capability to help keep a child's attention. Animation and sound routines are also utilized as positive reinforcement for correct answers.

The only problem I have found with Early Games is that, for some reason, Springboard has decided to copy protect it. The main menu and the games are all designed to allow small children to use the computer unsupervised if necessary. I fail to see the logic in giving any small child an original \$39.95 disk and expect him (or her) not to damage it. Thus I decided to make a backup copy for routine use.

The Investigation

When I first booted the Early Games disk it sounded as if the disk drive was constantly moving back to check a particular track for a nibble count. Since I had had success backing up other nibble counts with some commercial copy programs, I decided to try my favorite copy routines first. My initial attempt to back up Early Games was with the Automatic Bit Copier in Copy II Plus, Version 5.4. Although Early Games appeared to copy properly, the backup would not boot. My second try utilized the default parameters on Locksmith 5.0, Revision F. The Locksmith copy did boot though it would not run properly and required the disk drive to recalibrate on numerous occasions. This was an unacceptable solution!

Not wanting to attempt a boot code trace yet, I started looking for a way to enter the monitor in an attempt to capture the Read Write Track Sector (RWTS) portion of DOS. To my surprise my first attempt worked! I booted the the Early Games disk and then immediately pressed **⌘C**. This stopped the boot process but did not yield the Applesoft prompt which I had hoped for. Then I pressed the RESET key twice in quick succession and was rewarded with the Applesoft prompt! To see if I had allowed the boot time to load the modified RWTS I tried to catalog the disk. Typing "CATALOG" only returned a beep, indicating that DOS had been modified, so I tried the alternative way of cataloging a disk, "CALL 42350". This resulted in a 24 file catalog scrolling up on my screen! It appeared that I would be able to capture the early games RWTS and thus be able to use the Newspaper Controller routine to deprotect the Early Games disk. But it didn't turn out to be quite that easy.

The Procedure

After loading the Early Games DOS, which I verified by CATALOGing the Early Games disk (it would not CATALOG a standard DOS 3.3 disk), I entered the monitor. This was accomplished with the following command:

CALL -151

Since most RWTS routines normally reside in memory between \$B800 and \$BFFF (those are hexadecimal numbers which equate to decimal 47104 and 49151) it was necessary to move that block of memory to another area in memory which would not be used during a warm boot. That way when a warm boot is done the modified RWTS will not be overwritten during the boot. The following monitor command will move the Early Games RWTS to a safe area in memory:

1900<B800.BFFF

That command moves (copies) the block of memory from \$B800 through \$BFFF down to lower memory starting at \$1900. The "M" at the end of the above command tells the monitor to "Move" a block of memory.

Next it was necessary to reload DOS 3.3 into memory without losing the modified RWTS. This was accomplished by doing a "warm boot" from the monitor. Before the command was given for a warm boot, the Early Games disk was removed from disk drive #1 and replaced with a DOS 3.3 disk which did not have a boot program (normally called "HELLO") on it. The warm boot was executed from the monitor by using the following command:

C600G

The "G" stands for "Go" which tells the monitor to execute the commands starting at memory location \$C600. These are the Read Only Memory (ROM) instructions which tell your Apple how to do a "warm boot."

Once the Applesoft prompt came up I entered the following command to save the modified RWTS:

**BSAVE RWTS.EARLY GAMES,
A\$1900,L\$800**

Next I initialized a disk with a fast DOS (such as Diversi-DOS or Pronto-DOS) and a boot program named "HELLO":

INIT HELLO

This DOS would replace the modified DOS on the Early Games backup disk. Then I deleted

the HELLO program on the newly initialized disk:

DELETE HELLO

Once I had completed all of my preparations, I entered the original Newszap Controller from COMPUTIST No. 22, page 21. I quickly discovered that it couldn't read the DOS tracks (tracks 0-2) and it wouldn't copy track \$A. I was not concerned with the DOS tracks because I wanted to replace them on the backup, but track \$A required some investigation. Not knowing what was on track \$A, I loaded a sector editor to examine the track. A quick look at it proved out my earlier suspicions that it was used for a nibble count routine. The track was full of \$FF bytes with several marking bytes scattered throughout. Now all I had to do was to rewrite the swap controller so that it ignored track \$A.

After developing a way to make Super IOB V1.5 skip track \$A I wrote the controller listed below with a word processor and saved it as a standard text file. I called the controller file "EARLY GAMES CONTROLLER." After saving the controller I loaded Super IOB V1.5 back into memory:

LOAD SUPER IOB 1.5

Then I executed the controller so that it loaded into memory with Super IOB 1.5:

EXEC EARLY GAMES CONTROLLER

Finally, I started the Super IOB program by typing "RUN." I did not select the format option since I had already initialized a disk for the backup copy. Super IOB 1.5 then copied the Early Games disk to a standard DOS 3.3 disk.

Fixes

Once the copy process has been completed, LOAD the HELLO program into memory and LIST it. The following lines can be deleted:

```
3 ONERR GOTO 1
30 PRINT D$"BRUN RST,SND2"
32 PRINT D$"BRUN STOPPER"
```

You can also delete RST.SND2 and STOPPER from the deprotected disk.

Though these changes are not critical to the operation of the deprotected disk, they do allow you to exit to an unmodified version of DOS 3.3 by pressing the RESET key. If you do not make these changes you will have to reboot DOS after quitting the games.

Another change which you can easily make will recover track \$A, the one used for the nibble count. This will provide you some additional free space on your backup disk to save your child's drawings on. First you initialize another disk with DOS 3.3 (or a fast DOS) which has a boot program called HELLO. Don't forget to DELETE the HELLO program again after initializing the disk! Then use a file copy program, such as FID, to copy the files over to the new disk. This procedure releases track \$A which was unused, but not available, on your original IOB produced backup.

Step by Step


1) Initialize a blank disk with DOS 3.3 or a fast DOS and DELETE the boot program.

INIT HELLO DELETE HELLO

2) Type in the Early Games Controller listed below and save it as a text file (called EARLY GAMES CONTROLLER).

3) Load the Early Games disk in disk drive #1 and boot it.

PR#6

4) Type  as soon as the disk starts to boot.

5) Press the Reset key twice quickly. This will cause the Applesoft prompt to appear on the screen.

6) Enter the monitor and move the Early Games RWTS.

CALL -151 1900<B800.BFFFM

7) Replace the Early Games disk with a DOS 3.3 disk with no HELLO program. Then execute a warm boot from the monitor.

C600G

8) Save the modified RWTS to disk.

BSAVE RWTS.EARLY GAMES, A\$1900, L\$800

If you CATALOG your DOS 3.3 disk you should now have ten sector binary file named RWTS.EARLY GAMES.

9) Load Super IOB 1.5 and install the controller.

LOAD SUPER IOB 1.5

EXEC EARLY GAMES CONTROLLER

10) RUN Super IOB 1.5 but do not select the format disk option.

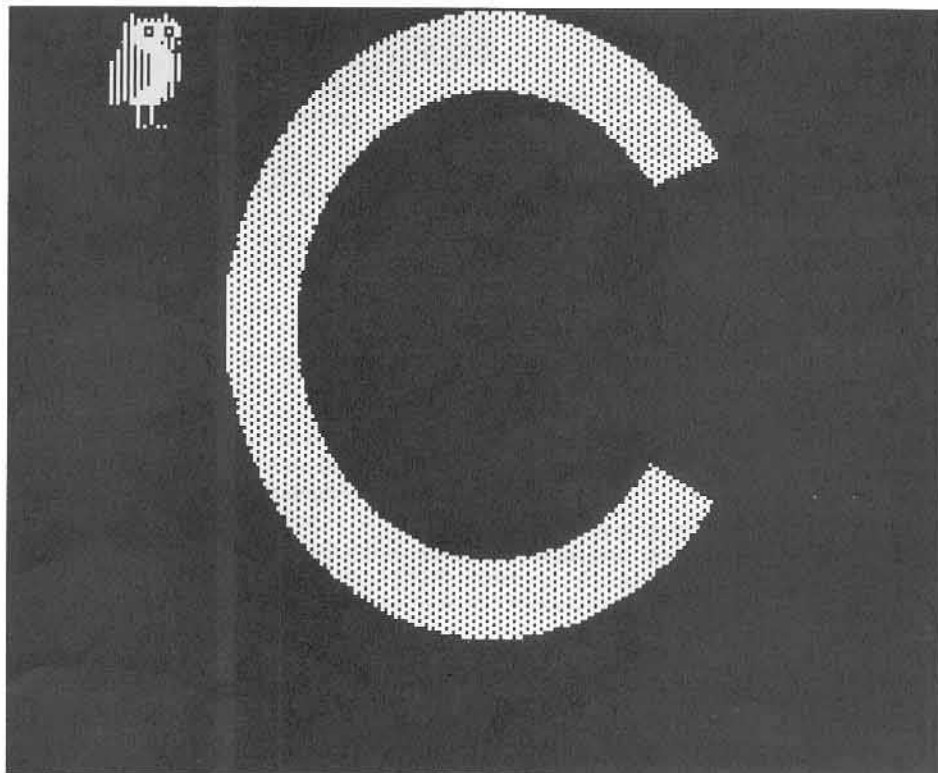
11) Make the recommended changes listed above, if desired, and let your children enjoy learning from a deprotected copy of Early Games for Young Children!

controller

```
1000 REM EARLY GAMES CONTROLLER
1005 XX = 0 : TK = 3 : LT = 10
1010 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 IF XX = 0 THEN TK = 11 : LT = 35 : XX = 1 : GOTO
1010
1060 HOME : PRINT "COPY^ DONE" : END
10010 PRINT CHR$(4) "BLOAD^ RWTS.EARLY^
GAMES,A$1900"
```

controller

1000	- \$356B	1040	- \$3BAC
1005	- \$8838	1050	- \$9EB0
1010	- \$D7D3	1060	- \$8559
1020	- \$3A9E	10010	- \$CFAC
1030	- \$080A		



Tawala's Last Redoubt

by Danny Pollak

Broderbund Software
17 Paul Drive
San Rafael, CA 94903

Requirements:

48K Apple][Plus or equivalent
Tawala's Last Redoubt
Super IOB v1.5
One blank disk

Tawala's Last Redoubt (TLR) is a game in which you play the part of the rebel leader, trying to defeat Emperor Tawala. You must use all of your resources to the best of your ability in order to overcome Tawala's forces and win the game.

The Protection

Considering when this game was published (1981), the protection is rather good. But then again, this is standard for Broderbund. They use multiple changing data headers as well as multiple address field closing marks. With the swap controller and the modified DOS 3.2 RWTS from the game, this program can be taken care of with Super IOB.

Step By Step

1) Initialize a blank disk with DOS (preferably a fast one) using "HELLO" as the hello program.

INIT HELLO

2) The next thing that must be done is to capture the foreign RWTS. To do this, boot the TLR disk. When the question "How quickly do you wish time to pass on Tawala's Last Redoubt?" appears, hit Reset until the drive stops.

3) Enter the monitor and move the RWTS (see note below) to a safe location.

CALL-151
1900<B600.BFFF

4) Insert a slave disk with no hello program, reboot normal DOS, and save the RWTS as a binary file on your Super IOB disk.



BSAVE RWTS.TLR,AS1900,LSA00

5) Install the controller below into Super IOB 1.5 and then run it. Tracks \$03 - \$22 will be copied and you will have a deprotected copy of TLR.

controller

```
60 LOMEM: 8960 : HIMEM: 9983 : GOTO 10010
360 POKE 253 , 25 : POKE 255 , 182 : POKE 224 , 10
      : CALL 832 : RETURN
1000 REM TLR CONTROLLER
1010 TK=3 : LT=35 : ST=12 : LS=12 : CD=WR : FAST
      = 1 : MB=130
1015 POKE 900 , 12
1020 GOSUB 490 : GOSUB 360 : 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 "COPYDONE" : END
10010 PRINT CHR$( 4 ) "BLOAD^ RWTS.TLR"
```

controller checksums

60	- \$E82F	1020	- \$56FF
360	- \$0B5C	1030	- \$646B
1000	- \$61E1	1040	- \$AE52
1010	- \$3B76	1050	- \$D5A8
1015	- \$09DC	10010	- \$9A69

Editor's note: The IOB controller and the instructions may look a bit strange here. This is because the controller has been set to copy DOS 3.2's 13 sectors per track (lines 1010 and 1015), and the RWTS of the game is slightly larger because it needs subroutines in the \$B600-\$B700 area. To fix that, we saved the RWTS plus the extra two pages and modified Super IOB to reserve extra room for them (line 60). Notice that room has to be taken from the buffer to do this. The swap routine (line 360) also needed modification to move more memory.

Technically, Beneath Apple DOS by Don Worth & Pieter Lechner considers the memory \$B600-\$B7FF as part of RWTS, but it normally contains non-critical information like the boot code (kept around for INITing).

Adventure Tips

ULTIMA III

- A single key and bribe can start you to unlimited, risk-free riches at Death Gulch.
- To ride through Death is much healthier than walking.

By Don R DeHalas

ULTIMA

- Buy ladder spells in the towns.
- Finding natural ladders in the dungeons is a very difficult business, and they are too hard to map.
- Stealing things in the towns is risky even for a high-level thief and to sell them back to the unsuspecting owners is outrageous, but it's a way to get needed funds and exotic advanced items before they are offered for sale.
- In the castle, kill the jester and get a key to unlock the princess.
- You'll have to fight some strong guards to do so, so have a lot of hit points. (Save before you enter castle.)
- In outer space, mapping the starbases will aid in getting through the scenario faster, later.

By Paul Wilson

SUSPECT

- Don't take the broken glass directly.
- Instead take the trash basket which Smythe put it in, and give that to the detective.
- Search the fairy mask for a clue.
- Meet Licia at the front door by being there at 9:18.
- Look up.
- Veronica wrote you a very nice letter of invitation.
- Why then was she so brusque to you when she went to the office at 9:01?
- Showing evidence to the detective.
- Being arrested at 11:40?
- Will thwart the frame-up.

By Paul Wilson

PRISONER II

- Find key.
- Look around maze.
- Go to school.
- Be good.
- Stay away from a bad element.
- Go to court, if you are thrown out go back a few times.
- Take a chance in court - write it down.
- The grail is forward.
- When you come to the end of the game until the truth then lie.
- The island is a computer.
- Do not give out your number, lie.

By Ed Higgins

It's the...

COMPUTIST Multi-Purpose Cloth

101 uses! Use it for...

Cleaning! Dusting! Polishing!
Shopping! Wiping Spills! Gardening!
A Birdcage Cover! A Birdcage Liner!
You can even wear it as clothing!

**A fashion statement!
A political statement!
An understatement!**

Yes, I want _____ C.M.P.C.s in the size(s) indicated below. I have enclosed \$9.95 plus tax and shipping for each t-shirt.

ADULT MENS: _____ Small _____ Medium _____ Large _____ X-Large

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

Exp. _____

Signature _____ CP32

Send check or money order to: CMPC PO Box 110937-A Tacoma, WA 98411
Washington orders add 7.8% sales tax. Foreign orders add 20% shipping and handling.
US funds drawn on US bank. OFFER GOOD WHILE SUPPLIES LAST.
NOT FOR ILLEGAL, IMMORAL, OR FATTENING USE.

68¢ per disk?

And that INCLUDES the Tyvek sleeve, too?

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

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

25
Verbatim disks
for only
\$17

- Sold only in sets of 25.
- 100% guaranteed.
- reinforced hubs.
- Tyvek sleeves & Write-protect tabs included.

Send your order to:
Disk Offer
SoftKey Publishing
PO Box 110937-A
Tacoma, WA 98411

Enclose \$3 shipping & handling for the first set PLUS \$1 for each additional set. Foreign orders add 20% shipping & handling US funds drawn on US bank. Washington state orders add 7.8% sales tax. Orders shipped via UPS, so please use street address.

Please send me _____ sets (25 disks per set) of 5 1/4" SS/DD diskettes. I have enclosed \$17 PLUS shipping & handling for each set and made checks / money orders payable to SoftKey Publishing.

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

Exp. _____

Signature _____

BACK UP YOUR DISKS

EDD Version 4 is the most powerful copy program available for backing up "uncopyable" or "copy-protected" disks. ■ In addition to backing up disks, **EDD 4** also features a hi-resolution graphic DISK SCAN option to help you locate information on a disk, a CERTIFY DISK option for certifying blank disks, and since it's very important that your disk drives are running properly (especially when copying disks), we have also included an EXAMINE DISK DRIVE option. ■ Even though **EDD 4** has been preset to copy the broadest range of copy-protections possible, **EDD 4** can be "modified" to back up almost any disk that runs on your Apple! ■ For the dedicated user, in addition to **EDD 4**, we are offering an **EDD 4 PLUS** version that includes a specially designed hardware card which allows **EDD** to copy EVERY bit of information from each track accurately! You can bet that if **EDD 4 PLUS** can't copy it, nothing will! ■ **EDD 4** runs on an Apple II, II Plus (including most compatibles, IIe, IIc, and III (using emulation mode), and is priced at **\$79.95**. ■ **EDD 4 PLUS** runs on Apple II, II Plus (including most compatibles), and IIe, and is priced at **\$129.95** (duodisk/unidisk 5.25 owners must add \$15 for a special cable adapter). Ask for **EDD** at your local dealer, or to order direct, include \$3 (\$6 foreign) shipping/handling for **EDD 4**, or include \$5 (\$8 foreign) for **EDD 4 PLUS**. ■ Mastercard and Visa accepted. All orders must be prepaid. ■ If you have an earlier version of **EDD**, you can update to **EDD 4** or **EDD 4 PLUS** at a reduced price. Send your **EDD** disk to us, and deduct \$50 from your order.

EDD is sold for the sole purpose of making archival copies ONLY!

UTILICO MICROWARE

3377 SOLANO AVENUE / SUITE 352 / NAPA, CA 94558 / 707-257-2420

ESSENTIAL DATA DUPLICATOR 4

Public Domain Software
SPECIAL
 From The COMPUTER LEARNING CENTER!

GET Over 170
**BUSINESS
 & FINANCE**
 Programs FOR ONLY
\$28.⁰⁰

- Forecast your investment profits, and evaluate the worth of your stock portfolio
- Experiment with Visicalc formulas, real estate and stock plots and design a business or personal letter writing format
- Organize your household expenses
- Balance your checkbook, manage your loans, payroll

The Business & Finance SPECIAL contains the following programs and many more...

Annuity	Real Estate Plot	Savings Growth
Bond Price & Interest	Visicalc Formulas	Treasury Bill Valuation
Budget Monthly	Business Finance	Address File
Decision Matrix	Check Stub	Compound Int. Tables
Financial Pak	Household Exp. Profile	File Manager
Keogh Savings Program	Income Tax 1040 For 77	Home Accounting
Mortgage Calculation	Inventory Company	Names Search
Regular Deposits	Average Growth Rate	File Cabinet I
Stocks	Check Book Balancer	Family Finance
Calendar Personal	Income Taxes	Gen Ledger Printer
Letter Writer	Interest Earned	Constar
Phone List	Loan Balance	etc...

ORDER NOW!

You will also receive a complete catalog of Public Domain Software available from the CLC with your order.

Yes! Please send me the Business & Finance Special Offer (Volumes 18-25 from the CLC Public Domain Library). I understand that Public Domain Software is not commercial quality and is supplied as-is and that orders are filled on double sided disks.

Name _____ S-4

Address _____

City _____ St _____ Zip _____

Country _____ Phone _____

VISA/MC _____ Exp _____

Signature _____

Send check or money order (US funds drawn on US bank) to: Computer Learning Center, PO Box 110876-HC, Tacoma, WA 98411. Washington residents add 7.8% sales tax. Foreign orders add 20% shipping and handling.

(The Computer Learning Center's PDS Library contains over 175 volumes, which will run on Apple II Plus computers and Apple compatibles. Most will also run on Apple IIe and IIc.)



Now-the ultimate back-up system!

**EDD 4[®] and
 TRAK STAR[™]**

**COMPLETE PACKAGE:
 159⁹⁵***
 Includes • TRAK STAR • 2-Drive Adaptor
 • EDD • Trak Star Patching Software

PRICED SEPARATELY:

TRAK STAR 99⁹⁵

2-Drive Adapter (required for 2-drive systems) \$12

Documentation: \$3 * Please add \$3 for shipping
 Refundable with the & handling. Foreign airmail &
 purchase of TRAK STAR handling, add \$8
 DuoDisk & IIc owners write.

**Save copying time
 with nibble programs**

- Works with nibble copy programs to display tracks and half tracks that the program accesses.
- Operates with any Apple®-compatible program.
- Save time by copying only the tracks being used.
- Displays up to 80 tracks and half-tracks; compatible with high density drives.
- If copied program doesn't run, Trak Star displays track to be recopied.
- Includes patching software for Trak Star.
- Compact size permits placement on top of disk drive.
- Does not use a slot in the Apple® computer.
- For Apple II, II+, IIe and compatibles.

Personal checks,
 M.O., VISA &
 Mastercard
 accepted.



Apple is a registered trademark of Apple Computer, Inc. EDD III is a trademark of Utilico Microware

Midwest Microsystems

To order, phone:
 913 676-7242

9071 Metcalf / Suite 124
 Overland Park, KS 66212


BACK ISSUES and LIBRARY DISKS

of COMPUTIST (formerly *Hardcore COMPUTIST*)
are still available, though some issues (marked NA) are sold out,
library disks are available for ALL issues of COMPUTIST.

*LIBRARY DISKS
are perfect companions for
COMPUTIST*

Documentation for Library Disks is in the corresponding issue.

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

Name _____ ID# _____
 Address _____
 City _____ State _____ Zip _____
 Country _____ Phone _____
 _____ Exp. _____
 Signature _____ CP32 _____

Send check or money order to: COMPUTIST PO Box 110937-A Tacoma, WA 98411. Most orders shipped UPS so please use street address. Offer good while supply lasts. In Washington state: add 7.8% sales tax.

Back Issue Rates For Foreign Orders

FOREIGN MAGAZINE ORDERS

Price for each magazine includes shipping.

1 - 2 copies 3 to 4 copies 5 or more copies

Canada/Mexico \$8.00 each \$7.00 each \$6.00 each

Other Foreign.....\$14.25 each.....\$13.25 each..... 12.25 each

FOREIGN DISK ORDERS

Disks are \$11.94 each (includes shipping). Special "Both" disk and magazine combinations shown do NOT apply to Foreign orders. US funds drawn on US banks. All foreign orders sent AIR RATES.

Issue	Mag \$4.75	Disk \$9.95	Both \$12.95
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 type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21.....	NA	<input type="checkbox"/>	NA
20.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
☆ 18..	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15.....	NA	<input type="checkbox"/>	NA
14.....	NA	<input type="checkbox"/>	NA
☆ 13..	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.....	NA	<input type="checkbox"/>	NA
☆ 11..	NA	<input type="checkbox"/>	NA
10.....	NA	<input type="checkbox"/>	NA
9.....	NA	<input type="checkbox"/>	NA
* 8.....	NA	<input type="checkbox"/>	NA
☆ * 7..	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.....	NA	<input type="checkbox"/>	NA
☆ * 4..	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
* 3.....	NA	<input type="checkbox"/>	NA
* Core 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
* 2.....	NA	<input type="checkbox"/>	NA
* 1.....	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
* Core 1	<input type="checkbox"/>	<input type="checkbox"/>	NA
Core 3.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computing 3	<input type="checkbox"/>	NA	NA

Special "Both" disk & magazine combination orders apply to one issue and its corresponding disk.

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

☆ We have a limited supply of these issues.

Description of Available Back Issues

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 Bard's 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 | Pistop II | Adventure 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 I | 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 |

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

22 *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's 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 | Arithmetickle | Arithmetickes 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 |

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

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

16 *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 I |

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

7 *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 I | Upper & Lower Case Output for Zork |

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

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 |

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 |

Back issues not listed
are no longer available.

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

Use the order form
on the other side of this page

Writer's Guide

COMPUTIST

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

MANUSCRIPT MECHANICS

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

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

TITLE OF WORK
FULL NAME OF AUTHOR
ADDRESS
PHONE NUMBER

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

The article and any accompanying program **SHOULD BE SUBMITTED AS A STANDARD TEXT FILE ON A DOS 3.3 DISK**. Label the disk with the title of the work and the author's full name and address. **ON DISK, TEXT MUST BE SINGLE-SPACED ONLY**. Please identify your editing program.

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

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

SUBJECTS

We prefer material on these topics:

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

WRITING YOUR ARTICLE

Observe the following points of style:

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

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

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

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

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

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

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

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

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

AUTHOR'S RIGHTS

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

PAYMENTS

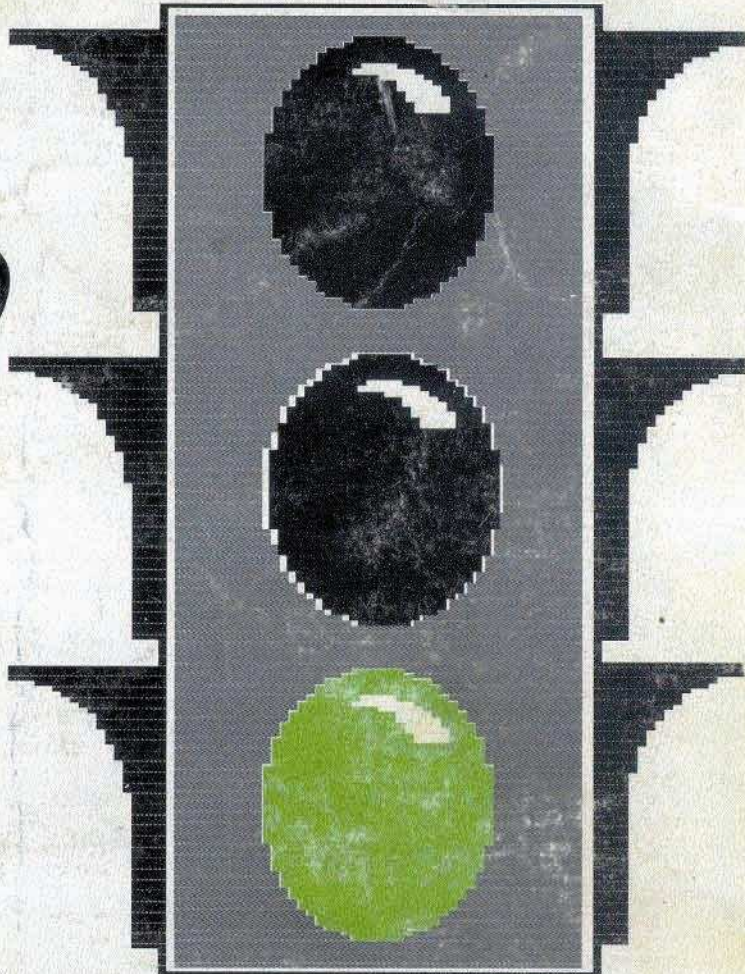
COMPUTIST pays upon publication. Rate of payment depends on the amount of editing required and the length of the article. Payment ranges from \$20 to \$50 per typeset page for an article. We also pay \$10 to \$20 for short softkeys and APT's. A fully explained softkey accompanied by the commercial disk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

Please mail your submissions to:

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

GO AHEAD

**It's legal
to make
backups
of your locked-up
commercial software**



The Book Of Softkeys

Shows you how to remove copy protection from commercial software.

Volume I (157 pages), contains softkeys for: Akalabeth, Ampermagic, Apple Galaxian, Aztec, Bag of Tricks, Bill Budge's Trilogy, Buzzard Bait, Cannonball Blitz, Casino, Data Reporter, Deadline, Disk Organizer II, Egbert II Communications Disk, Hard Hat Mack, Home Accountant, Homeword, Lancaster, Magic Window II, Multi-disk Catalog, Multiplan, Pest Patrol, Prisoner II, Sammy Lightfoot, Screen Writer II, Sneakers, Spy's Demise, Starcross, Suspended, Ultima II, Visifile, Visiplot, Visitrend, Witness, Wizardry, Zork I, Zork II, Zork III, plus how-to articles and program listings of need-to-have programs used to make unprotected backups.

Volumes II and III are being compiled now!

YES, I want ____ copies of The Book of Softkeys Volume 1. I have enclosed \$20 per book. Foreign orders add 20%. U.S. funds drawn on U.S. banks. Washington State orders add 7.8% sales tax. Send your orders to:

SoftKey Publishing
PO Box 110937-A
Tacoma, WA 98411

Name _____ ID# _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

  _____ Exp. _____

Signature _____ CP32