

# COMPUTIST

Issue # 50

December 1987

USA \$3.75

Canada & Mexico \$7.00

All Others \$8.75

## Features:

- M.A. Horton's  
**APT Writer for...**  
*Super Boulderdash*
- William Green's  
**Softkey to...**  
**Activision software**
- Thomas Beach's  
**Double Your F-8 ROM Space**  
**Without Motherboard Surgery**
- Jack Moravetz's  
**Softkey to...**  
**MECC software**
- Ron Balewski's  
**Ace-Apple Bimodal Switch:**  
**Two machines in one**
- A. L. Head's...  
**Softkey to...**  
**PFS ProDOS software**
- Using Sider hard drives, 3 1/2" 800K. drives, & 5 1/4" drives in a DOS 3.3 Environment**

The Learning Company™  
by Warren Robinett  
and Leslie Grimm  
© 1981, 1985  
The Learning Company  
version 4.0  
sensors  
targets  
ROCKY'S BOOTS™

## Softkeys:

♣ Aliens, ♣ Alter Ego, ♣ Alternate Reality, ♣ Amazing Reading Machines, ♣ Amazon, ♣ American Challenge, ♣ Arcade Album # 1, ♣ Arithmetic Critters, ♣ Award Maker, ♣ Baseball Database, ♣ Bard's Tale II: Destiny Knight, ♣ BC's Quest for Tires, ♣ Bop & Wrestle, ♣ Championship Boxing, ♣ Championship Wrestling, ♣ Clock Works, ♣ Commando, ♣ Computer Preparation for SAT, ♣ Conflict In Vietnam, ♣ Counting Critters, ♣ Crisis Mountain, ♣ MECC Dataquest: Fifty States, ♣ Deluxe Paint II, ♣ Dino Eggs, ♣ Disney Card & Party Shop, ♣ Disney Comic Strip Maker, ♣ Draw Plus, ♣ Eidolon, ♣ Electric Crayon ABCs, ♣ Expedition Amazon, ♣ Facemaker, ♣ First Letter Fun, ♣ Fish Scales, ♣ Fun From A to Z, ♣ Game Maker, ♣ GBA Championship Basketball, ♣ GFL Championship Football, ♣ Graphicwriter I, OR & I, IR, ♣ Great Road Race, ♣ Hacker II, ♣ Hardball, ♣ Infiltrator II, ♣ Instant Music, ♣ James Bond 007: A View To A Kill, ♣ Jenny's Journeys, ♣ Kung Fu Master, ♣ Little People, ♣ List Handler, ♣ Manic Mansion, ♣ Mastery Arithmetic Games, ♣ Market Place, ♣ Master of the Lamp, ♣ Math Rabbit, ♣ Microzine #17, ♣ Might and Magic, ♣ Mission In Our Solar System, ♣ Moebius, ♣ Music Construction Set, ♣ Music Studio, ♣ Number Munchers, ♣ Paint With Words, ♣ Paintworks Plus, ♣ Path Tactics, ♣ pfs:File, ♣ pfs:Graph, ♣ pfs:Plan, ♣ pfs:Report, ♣ pfs:Write, ♣ Phonics Prime Time, ♣ Portal, ♣ Principal's Assistant, ♣ Print Shop ProDOS 8 v1.4, ♣ Print Shop Graphics Holiday Edition, ♣ Quickflash!, ♣ Reader Rabbit, ♣ Realm of Impossibility, ♣ Robot Odyssey I v.2.0, ♣ Rocky Horror Show, ♣ Rocky's Boots v4.0, ♣ Saracen, ♣ Shanghai, ♣ Silent service, ♣ Skylab, ♣ Sound Tracks, ♣ Speedy Math, ♣ Spindizzy, ♣ Street Sports Baseball, ♣ Sub-Mission, ♣ Super Boulderdash, ♣ Tass Times in Tonetown, ♣ Thexder, ♣ Top Fuel Eliminator, ♣ Word Handler, ♣ Word Munchers, ♣ Words at Work, ♣ World Karate Championship, ♣ Writer's Choice: Elite, ♣ Zardax v5.2.1 ... ♣ Apple IIgs softkey.

COMPUTIST  
PO Box 110846-T  
Tacoma, WA 98411

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

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

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

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

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

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

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

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

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

*United States Code title 17, §117*

## **COMPUTIST**

**Publisher-Editor:** Charles R. Haight  
**Graphics Editor:** Ryuji  
**Circulation:** Karen Fitzpatrick  
**Advertising:** (206) 474-5750  
**Publishing:** SoftKey  
**Printing:** Valco Printing

- Address all advertising inquiries to:  
COMPUTIST  
Advertising Department  
PO Box 110816  
Tacoma, WA 98411
  - Mail manuscripts or letters to:  
COMPUTIST  
PO Box 110846-K  
Tacoma, WA 98411
  - Unsolicited manuscripts are assumed to be submitted for publication with all and exclusive rights belonging to COMPUTIST. For more information on submitting manuscripts, consult the writer's guide on the inside of the back cover.
  - Entire contents copyright 1987 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 volunteer and paid 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 12 issues):  
U.S.—\$32                      U.S. 1st Class—\$45  
Canada & Mexico—\$45 Other Foreign—\$75
- Direct subscription inquiries to:  
COMPUTIST  
Subscription Department  
PO Box 110846-T  
Tacoma, WA 98411
  - **DOMESTIC DEALER RATES:**  
Call (206) 474-5750 for more information.
  - **Change Of Address:** Please allow 4 weeks for change of address to take effect.
  - On postal form 3576 supply your new address and your most recent address label.
  - Issues missed due to non-receipt of change of address may be acquired at the regular back issue rate.

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

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

**Time to renew?**

- Check your mailing label to see if you need to renew your subscription.
- Use the order form on this page to renew, and don't forget to send your most current mailing label to help expedite your renewal

**Moving soon?**

- If you're moving, let us know at least 30 days in advance.
- Issues missed due to non-receipt of Change-of-Address may be acquired at the regular back issue rates.
- Remember, the Post Office does not forward third class mail unless requested. COMPUTIST is not responsible for replacing issues lost while forwarding order is in effect.

**New Year's  
RESOLUTION(s)**



*Subscribe*

*to*

*Computist*

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

- I am...  A new Subscriber  
 Renewing my current subscription  
 Changing my address (please include last mail label)

- Subscription type:  
 U.S. - \$32  
 U.S./Canada/Mexico First Class - \$45  
 U.S./Canada/Mexico First Class plus Library Disk - \$100  
 All other Foreign - \$75  
 All other Foreign plus Library Disk - \$140

Name \_\_\_\_\_ ID# \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Country \_\_\_\_\_ Phone \_\_\_\_\_

 \_\_\_\_\_ Exp. \_\_\_\_\_

Signature \_\_\_\_\_ CP50

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

David G. Alexander	21
Ron Balewski	25
Bill Beach	11
Thomas Beach	17
Jack Bernhardt	10
T. F. Blake	37
Elvin Chan	17
Computist In Colorado	22
The Dabbler	10
Philip Doherty	23
A. E.	16
John Einstein	11
Matt Evans	12
Dennis Gaunt	15
William Green	16
The Hacksaw	16
A. L. Head, Jr.	26
Magnus Hedlund	10
Daryl Higa	14
Michael A. Horton	7
Kracker Jack	9
Guy Kendall	9
Mr. E. Lau	18
Dick Lavellee	22
S. T. M.	23
Mike Maginnis	9
Willem Moolenaar	36
Jack Moravetz	23
Tamaki Murakami	22
Roy Newberry	19
Jack R. Nissel	12
Robert Paquette	17
Scuzzy Port	19
William O. Romine	37
Bill Rossmann	17
Lenny Schmidt	18
J. Slate	23
Brian Symonds	10
Edward Teach	38
Jerome Thelia	12
Mathew J. Tomczik	21
Brian A. Troha	19
Peter Wilding	16

# Editorial.....

Judging by the letters I've been reading, there is still a lot of confusion about RDEX both here and with our readers. I'm going to answer some of your questions here.

■ **Letters to authors:** We do not print our writer's addresses unless they specifically request it. This is to protect them from the idiots that would waste their valuable time and from the crazies that would do worse. Our writers are entitled to their privacy. The unfortunate side-result is that reader correspondence is also blocked. There is a way for you to contact an author if we have their address. Write your letter and put it in an envelope addressed to the authors name with your full return address and correct postage. Place this envelope into another envelope and send it to us. We will put the correct address on the envelope (providing we have one) and mail it from here.

■ **Identical softkeys:** We receive a lot of softkeys, some of which look identical. If upon closer examination there are differences, even minor ones, we will print all versions. If we do not have the program in question, it is impossible to second guess the author. If one softkey has one more edit than another, we don't know if the first softkey is incomplete or if the second softkey has an extra edit or even if the two softkeys are for different versions. It's better to print them both and let you decide.

While we're on the subject of softkeys I'd like to request that you include the hex bytes before and after the edits as a search string for those readers who may have a slightly different version.

■ **BoulderDash II:** Before anyone else jumps on my case... NO, I didn't know that *Super BoulderDash* is the same as *BoulderDash II*. And yes, we published a very complete softkey for *BoulderDash I & BoulderDash II* in *COMPUTIST #38*.

■ **Pictures:** When sending softkeys to RDEX on disk, please include hi-res pictures files from that program whenever possible. We need something for our covers and some may find their way inside.

■ **KUDOs:** Reader recommendations for quality software, hardware and exceptional service are invaluable to other readers. If you have a program that you feel ranks right up there with sex and fine chocolate, let the other readers know. Also, if you've received something from the wrong end of the horse, let us know so others won't have to repeat your experience. One condition: reader recommendations must have a real name and address. We can't print recommendations from anonymous sources.

■ **Starter Kit Documentation:** No, there isn't any easy way to print the "DOC" files. I suggest you load them into a text editor, delete the control characters and reformat the files, then use the editor to make your hard copies.

■ **Previous Submissions:** I still have a lot of catch-up work in the writing letters department. There's a pile of submissions still to be sorted, including some that were accepted before our editorial reorganization. If you submitted an article and have not received a letter from the RDEX editor, there's a good chance your material is still in that pile. I'm writing letters as fast as I can but there are so many folders and each requires a different letter. **You can shorten this process by sending a letter to RDEX naming your submission and giving us permission to use your material in RDEX.** You could probably make it into the next issue.

■ **Letters to COMPUTIST:** If you send material to RDEX make sure it's addressed to "RDEX". Letters sent to RDEX are uploaded first. Other letters have to be evaluated and if the intent is not clear they are put into another "box" until a follow-up letter can be written requesting clarification. And you know how far behind I am in my letter writing.

## ■ BUGS

■ **COMPUTIST #49, Science Toolkit, page 27.** Line 1020 and line 1040 of the controller should read:

```
1020 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST < 9 THEN 1020
1040 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST < 15 THEN 1040
```

# COMPUTIST

## Table of Contents

# #50

### Softkeys:

Aliens.....	23, 37
Alter Ego.....	23
Alternate Reality.....	8
Amazing Reading Machines.....	24
Amazon.....	22
American Challenge.....	11, 39
Arcade Album #1.....	12
Arithmetic Critters.....	24
Award Maker.....	10
Baseball Database.....	19
Bard's Tale II: Destiny Knight.....	37
*Bard's Tale IIgs.....	23
BC's Quest for Tires.....	16
Bop & Wrestle.....	17, 39
Championship Boxing.....	40
Championship Wrestling.....	36
Clock Works.....	24
Commando.....	36
Computer Preparation for SAT.....	14
Conflict In Vietnam.....	11
Counting Critters.....	24
Crisis Mountain.....	16
Dataquest: Fifty States.....	24
*Deluxe Paint II.....	19
Dino Eggs.....	16
Disney Card & Party Shop.....	18, 33
Disney Comic Strip Maker.....	18
*Draw Plus.....	21, 23
Eidolon.....	38
Electric Crayon ABCs.....	25
Expedition Amazon.....	12
Facemaker.....	40
First Letter Fun.....	24
Fish Scales.....	24
Fun From A to Z.....	24
Game Maker.....	17, 21
GBA Championship Basketball.....	37
GFL Championship Football.....	17, 37
*Graphicwriter I OR & I IR.....	20
Great Road Race.....	17
Hacker II.....	17, 21
Hardball.....	14
Infiltrator II.....	36
*Instant Music.....	19
James Bond 007: A View To A Kill.....	9
Jenny's Journeys.....	24
Kung Fu Master.....	40
Little People.....	17
List Handler.....	26
Manic Mansion.....	37
Market Place.....	24
Master of the Lamp.....	17
Mastery Arithmetic Games.....	19
Math Rabbit.....	9
Microzine #17.....	21
Might and Magic.....	10, 12
Mission In Our Solar System.....	15
Moebius.....	15
*Music Construction Set.....	9, 21
Music Studio.....	9
Number Munchers.....	24
Paint With Words.....	24
*Paintworks Plus.....	9
Path Tactics.....	24
pfs:File.....	28
pfs:Graph.....	28
pfs:Plan.....	28
pfs:Report.....	28
pfs:Write.....	28
Phonics Prime Time.....	24
Portal.....	36
Principal's Assistant.....	10
*Print Shop ProDOS 8 v1.4.....	22, 23
Print Shop Graphics Holiday Edition.....	19
Quickflash!.....	24
*Reader Rabbit.....	9, 23
Realm of Impossibility.....	20
Robot Odyssey I v.2.0.....	22
Rocky Horror Show.....	17
Rocky's Boots v4.0.....	32
Saracen.....	39
*Shanghai.....	9, 21
Silent Service.....	30
Skylab.....	24
Sound Tracks.....	24
Speedy Math.....	24
Spindizzy.....	12, 17
Street Sports Baseball.....	36, 40
Sub-Mission.....	39
Super Boulderdash.....	26
Tass Times in Tonetown.....	37
*Thexder.....	19
Top Fuel Eliminator.....	39
Word Handler.....	26
Word Munchers.....	24
Words at Work.....	24
World Karate Championship.....	14
*Writer's Choice Elite.....	23
Zardax v5.2.1.....	130

### A.P.T.s:

Autoduel.....	9
Black Magic.....	7
Commando.....	36
Expedition Amazon.....	12
Marble Madness.....	7
Rescue Raiders.....	8, 14
Sub-Mission.....	40
Super Boulderdash.....	7
Sword of Kadash.....	8

### Playing Tips:

Alternate Reality.....	16
Beyond Castle Wolfenstein.....	15, 16, 18
Coveted Mirror.....	10
Random.....	15
Ultima IV.....	10
World Karate Championship.....	14

**Editorial.....4**

**Bugs.....4**

**Most Wanted Softkeys.....40**

### Features:

APT Writer for...  
**Super Boulderdash.....7**

M.A. Horton's program automatically modifies *Super Boulderdash* so that you can get extra men or move on to the next cave any time you want.

**Activision software.....16**

William Green's quick-n-easy softkeys for: *Game Maker*, *Hacker II*, *Little Computer People*, *Great American Road Race*, *Spindizzy*, *Master of the Lamps*, *Rocky Horror Show*.

**Double Your F-8 ROM Space Without Motherboard Surgery.....17**

Thomas Beach's hardware project doesn't permanently alter your motherboard.

**MECC software.....23**

Jack Moravetz's Super IOB controller will completely deprotect *Arithmetic Critters*, *Counting Critters*, *Clock Works*, *First Letter Fun*, *Fun From A to Z*, *Jenny's Journey*, *Dataquest Fifty States*, *Number Munchers*, *Paint with Words*, *Path Tactics*, *Phonics prime Time*, *Quickflash!*, *Skylab*, *Sound Tracks*, *Speedway Math*, *Amazing Reading Machines*, *Market Place*, *Word Munchers*, *Words at Work*.

**Ace-Apple Bimodal Switch: Two machines in one.....25**

Ron Balewski's hardware project will give your Franklin Ace 1000 complete Apple-compatibility (or make your Apple completely Ace-compatible) with the flick of a switch

**PFS ProDOS software.....28**

A. L. Head's key to unlocking Software Publishing Corporation's pfs ProDOS series: *pfs:Write*, *pfs:File*, *pfs:Graph*, *pfs:Report*, *pfs:Plan*.

**Using Sider hard drives, 3 1/2" 800K drives, & 5 1/4" drives in a DOS 3.3 Environment.....34**

A. L. Head puts them all under DOS 3.3.



# Readers Data

For serious users of Apple computers: II, II+, //c, //e, IIGs, etc...

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

## What is a softkey, anyway?

Softkey is a term which we coined to describe a procedure that removes, or at least circumvents, any copy-protection on a particular disk. Once a softkey procedure has been performed, the resulting backup copy can usually be copied by the normal copy programs (for example: COPYA, on their DOS 3.3 System Master disk).

## Commands and control keys

In any article appearing in COMPUTIST, commands which a reader is required to perform are set apart by being in boldface and on a separate line:

### PR# 6

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

Control characters are specially boxed:

**6** **Ⓜ**

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

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

## Special requirements

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

## Software recommendations

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

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

✓ *COPYA*, *FID* and *MUFFIN* from the *DOS 3.3 System Master* disk are also useful.

## Super IOB and Controllers

This powerful deprotection utility (*COMPUTIST Starter Kit*) and its various Controllers are used in many softkeys. (It is also on each *Super IOB Collection* disk.)

## Reset into the Monitor

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

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

**Apple II+, compatibles:** 1) Install an F8 ROM with a modified reset-vector on the computer's motherboard as detailed in the *Modified ROM's* article (*COMPUTIST #6* or *Book Of Softkeys III*) or the *Dual ROM's* article (*COMPUTIST #19*).

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

## Recommended literature:

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

## Keying in Applesoft programs:

BASIC programs are printed in a format that is designed to minimize errors for readers who key in these programs. If you type:

**10HOME:REMCLEAR SCREEN**

The LIST will look like:

10 HOME : REM CLEAR SCREEN

...because Applesoft inserts spaces into a program listing before and after every command

word or mathematical operator. These spaces usually don't pose a problem except in line numbers which contain **REM** or **DATA** commands. There are two types of spaces: those that have to be keyed and those that don't. Spaces that must be typed appear in *COMPUTIST* as delta characters ( $\delta$ ). All other spaces are there for easier reading. NOTE: If you want your checksums (See **Computing checksums**) to match up, you must key ONLY the  $\delta$  spaces after **DATA** statements.

## Keying In Hexdumps

Machine language programs are printed in *COMPUTIST* as hexdumps, sometimes also as source code. Hexdumps are the shortest and easiest format to type in. You must first enter the monitor:

### CALL -151

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

When finished, return to BASIC with:

### 3DOG

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

## Keying in source code

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

## Computing checksums

Checksums are 4-digit hexadecimal numbers which tell if you keyed a program exactly as it appears in *COMPUTIST*.

There are two types of checksums: one created by the *CHECKBIN* program (for machine language programs) and the other created by the *CHECKSOFT* program (for BASIC programs). Both appeared in *COMPUTIST #1* and *The Best of Hardcore Computing*. An update to *CHECKSOFT* appeared in *COMPUTIST #18*.

If the published checksums accompanying program listings and hexdumps do not match those created by your computer, then you typed the program incorrectly. The line where the first checksum differs has an error.

# EXchange



who want all their software backed up and COPYA-able

## CHECKSOFT instructions:

LOAD filename  
BRUN CHECKSOFT

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

## CHECKBIN instructions:

CALL -151  
BLOAD filename

Install CHECKBIN at an out of the way place  
BRUN CHECKBIN, A\$6000

Get the checksums by typing the Starting address, a period and the Ending address of the file followed by a **[&Y]** **[RETURN]**.

SSS.EEE **[&Y]**

Correct the lines at which the checksums differ.



when  
writing  
a...

## letter to RDEXed

- Remember that your letters or parts of them may be used in the new Readers Data Exchange even if you don't address it to the RDEX editor. Correspondence that gets published may be edited for clarity, grammar and space requirements.

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

- Address your letters to:

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

- Although COMPUTIST can no longer purchase short softkeys and articles, please continue to contribute them but place them in a letter to the editor so that they get published in the RDEX as soon as possible.

-RDEXed

Michael A. Horton

† A.P.T for...

Black Magic  
Datasoft

## Infinite Arrows & Spells

I would like to say thanks to John Wiegley for the info on backing up *Black Magic*. In return, here is an APT for it.

Set your sector-editor as said in the article in COMPUTIST #48, so that it can read track \$1, sector \$0. Then make the following sector-edits. The first is for infinite arrows and the second is for infinite spells.

Trk	Sct	Bytes	From	To
\$05	\$00	\$F9-	C6 E9	EA EA
\$05	\$06	\$3D-	C6 E7	EA EA

**Playing Tip...** You must collect at least 1 arrow or 1 spell to use arrows or spells respectively.

If your copy does not have the same values at the same location then scan for the 'From' values and change them to the 'To' values.

A.P.T for...

Marble Madness  
Electronic Arts

## Unlimited Time

Here is a helping hand for the *Marble Madness* fan. In order to have unlimited time do the following sector-edit:

Trk	Sct	Bytes	From	To
\$0A	\$0F	\$20	08	00

If you don't find the 'from' value at that location scan the disk for BD 77 60 F0 08 A9 02. Change the 08 to 00.

## You're NOT history

You may also run into the "You're history!" problem. The following sector-edits will fix that but it will also create a problem...

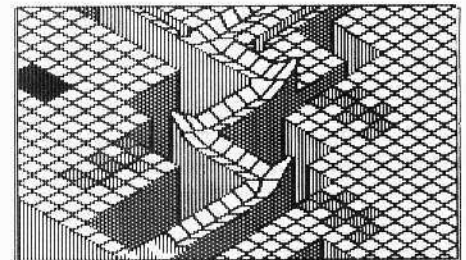
**Warning:** If you 'die' it will restart your marble where you died.

Trk	Sct	Bytes	From	To
\$0A	\$03	\$73-	9D 77 60	EA EA EA
		\$82	20	60

If you don't find the 'from' value at that location, scan the disk for A9 00 9D 77 60 A9 07. Change 9D 77 60 to EA EA EA. The sequence of 20 66 FD should follow this. Change 20 to 60.

## Entrance to the secret Maze

The entrance to the secret maze is on the first level. On the following picture there is a black trapezoid on the left part of the picture. This is where you must be when there are 13 seconds left on your timer if you wish to get to the secret maze. If you don't want to...



A.P.T for...

Super Boulderdash  
Electronic Arts

The following program is for all you *Super Boulderdash* players. It modifies sectors on your back-up disk using a 'data table', so that the mod fits the side of the disk that you choose. In order to use this, make a backup copy of the original (COMPUTIST #38) then type in the program and save it on a DOS 3.3 disk.

**BSAVE SUPER BOULDERDASH APT, A\$800,  
L\$168**

To use the APT:

- 1 Run the program:  
**BRUN SUPER BOULDERDASH APT**
- 2 Insert the game backup disk.
- 3 Select the proper side.

In order to use the APT while playing, do whichever of the following that you wish in order to use it.



hold down	then press	What it does
Button 0	Esc	Adds 1 to # of men
Button 1	Esc	Advance to next cave

Once in a great while the advance to the next cave does not work. If it doesn't work the first time, try it again and it should work properly. Also, the **APT WRITER** won't tell you if there are any errors while trying to access your backup copy.

## APT WRITER

```
0800: 20 2F FB 20 58 FC A0 20 $B244
0808: B9 1E 09 99 00 06 B9 42 $91D4
0810: 09 99 00 07 88 D0 F1 8D $6483
0818: 10 C0 AD 00 C0 10 FB C9 $5ADD
0820: C2 F0 0A C9 C6 D0 F0 A9 $2D57
0828: 00 A2 00 F0 04 A9 0C A2 $7E35
0830: 0D 85 FF 86 FE A9 00 85 $717C
0838: FD 8D EB B7 8D F0 B7 A9 $0077
0840: 20 85 01 8D F1 B7 A4 FE $930F
0848: B9 AA 08 F0 B3 8D EC B7 $421D
```

```
0850: C8 B9 AA 08 8D ED B7 A9 $C517
0858: 01 8D F4 B7 20 A3 08 A4 $7004
0860: FE C8 C8 B9 AA 08 85 00 $0F75
0868: C8 B9 AA 08 85 02 A0 00 $7DB5
0870: A6 FD BD C0 08 C9 FF F0 $4341
0878: 1C 91 00 E8 C8 C4 02 90 $551F
0880: F1 86 FD A5 FE 18 69 04 $01F9
0888: 85 FE A9 02 8D F4 B7 20 $D49F
0890: A3 08 4C 46 08 86 FD A6 $1BE8
0898: FF BD 07 09 E6 FF A6 FD $29C8
```

```
08A0: 4C 79 08 20 E3 03 20 D9 $A3FB
08A8: 03 60 0E 08 28 03 0D 0E $26FD
08B0: FE 02 0D 0D 00 42 00 0E $92B1
08B8: 07 E7 03 0D 0D 38 44 00 $86CD
08C0: 4C FF 0A A9 20 8D FF 0A $37F0
08C8: A9 40 8D FF 0A A0 00 A9 $F44F
08D0: 00 99 00 20 99 00 40 C8 $99C0
08D8: D0 F7 EE FF 0A EE FF 0A $CE65
08E0: AD FF 0A C9 60 D0 E8 AD $38A9
08E8: 54 C0 60 AD 61 C0 10 04 $AF68
```

```
08F0: A6 26 F6 28 AD 62 C0 10 $0111
08F8: 03 4C FF FF AD 61 C0 10 $7529
0900: 03 4C FF FF 4C FF FF 26 $D643
0908: 0E 11 0E 11 11 20 18 E7 $19A9
0910: 16 18 18 60 48 4B 48 4B $8823
0918: 4B DF 19 A6 18 D7 19 C9 $A205
0920: CE D3 C5 D2 D4 A0 D3 D5 $6E3B
0928: D0 C5 D2 A0 C2 CF D5 CC $EA0B
0930: C4 C5 D2 C4 C1 D3 C8 A0 $15F1
0938: C2 C1 C3 CB AD D5 D0 A0 $D248
```

```
0940: CE CF D7 C3 C8 C5 C1 D4 $2D80
0948: A0 06 D2 CF CE D4 A0 D3 $F7B1
0950: C9 C4 C5 A0 CF D2 A0 02 $FA22
0958: C1 C3 CB A0 D3 C9 C4 C5 $43B2
0960: BF A0 A0 A0 A0 A0 00 00 $3C33
```

Softkey for...

## Alternate Realty—The Dungeon

Datasoft

Here is a controller to deprotect *Alternate Realty: The Dungeon*. This is for **Disk 1, Side 1 of the Dungeon Disk**. In order to copy the other sides use any quick whole-disk copier.

### Controller

```
1000 REM CON : A R DUNGEON
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 GOSUB 300
1060 HOME : PRINT "COPY*DONE" : END
1070 DATA 0.5.73.0
```

### Controller Checksums

```
1000 - $356B 1040 - $A1C7
1010 - $2544 1050 - $7BCE
1020 - $08CD 1060 - $08ED
1030 - $0B00 1070 - $B813
```

If this does not work scan the backup disk for C9 CC D0 ED BD 8C C0 and change ED to 00.

More A.P.T.s for...

## Sword Of Kadash

Penguin

Those APT's sent in by Russell Meyers are okay, but what would you choose: lots... or infinite. His *Sword of Kadash* A.P.T. is just so-so. I say this because 'why should you have to die at all?'

### Immortality

Scan for E9 01 8D 0D 44 B0 06 D8 68 and change the last 2 bytes from D8 68 to 90 04.

Your hit points go down but you never die.

### Infinite spells

Scan for AD 15 44 F0 1C AD FF 6E and change F0 1C to EA EA. Scan for 8D FF 17 AD 15 44 F0 3E and change F0 3E to EA EA.

### No more illegal character disk

Scan for 20 00 03 AD 16 44 D0 03 and change the 3 bytes following it to EAEAEA. Scan for 90 F8 CD 19 44 F0 03 and change the 3 bytes after it to EA EA EA.

**Playing Hint:** Sometimes secret doors are blocked by shootable invisible walls.

## What's What On Your Character Disk

Here is a list to show you what's what on your character disk. All of the data is on track \$22, sector \$0F.

Byte	Explanation
00	your room #(\$00-\$FF)
01	integer X-coordinate divided by 7
02	Y-coordinate
03	X-coord. remainder (X = BS01*7 + BS03)
04	weapon #(\$00-\$0B), \$FF if cursed
05	armor #(\$0C-\$12), \$FF if cursed
06	weapon # if cursed
07	armor # if cursed
08	level
09	are you cursed? \$00 = NO \$01 = YES
0A	low byte of maximum hit points XXYY
0B	high byte of maximum hit points XXYY
0C	low byte of current hit points XXYY
0D	high byte of current hit points XXYY
0E	armor class
0F	???
10	low byte of experience XXYY
11	high byte of experience XXYY
15	number of spells
16	usually \$FF, must be non-zero for character to work
19	(\$00-\$15 EORed together)
1A	number of lives left

A.P.T. for...

## Rescue Raiders

Sir-tech

### Unlimited bullets and napalm.

Trk	Sct	Bytes	From	To
\$0F	\$04	\$26	DE F6 60	EA EA EA

### Unlimited bombs

Trk	Sct	Bytes	From	To
\$0F	\$04	\$9C	DE F4 60	EA EA EA

### Unlimited missiles

Trk	Sct	Bytes	From	To
\$0F	\$06	\$37	DE 02 61	EA EA EA

### Turn Bullets into napalm

Trk	Sct	Bytes	From	To
\$0F	\$04	\$2D	F0 E3	EA EA

**Playing Hint:** Missiles fire when at the same altitude as enemy copter.





**Mike Maginnis**

A.P.T. for...

**Auto-duel**  
Broderbund

### Ressurrect a dead Auto-dueler

Mr. Scott, in his **Auto-duel Editor** in **COMPUTIST #42**, page 22 (which is excellent - I have used it countless times since it was printed), failed to mention how to resurrect a character if it dies. Do you know how frustrating it is if a character that you've been working on for months dies and you have to start over?

Well, to resurrect a dead Auto-dueler, simply change byte \$1B of track \$1A, sector \$00 to 80 with a sector-editor and he/she lives again!

**Kracker Jack**

\*Iigs Softkey for...

**Music Construction Set (MCS)**  
Electronic Arts

**1** Copy the MCS program to a blank 3 1/2" disk. (I use *Diversi-Copy*). It will copy without errors, but it won't run.

**2** Format a blank 5 1/4" disk with ProDos (I use *Copy II Plus 7.4*)

**3** Copy the file *MUSICGS.SYSTEM* to the formatted floppy.

**4** With a sector-editor (*Copy II Plus* again), look for the sequence 20 1A B5 A9 01 in the file. I found it on track \$9, sector \$1, byte \$31. Change 20 to 60 and write it back to the disk.

**5** Copy the file back to your 3 1/2" copy of MCS and you are done!

\*Iigs Softkey for...

**Paintworks Plus**  
Activision

**1** Make a copy of your 3 1/2" original disk. Ignore any read errors on block 7. I use *Diversi-Copy*. It's the fastest copier that I have found for 3 1/2" disks.

**2** Boot your Apple Iigs system disk and get into BASIC.

**3** Type:

**PREFIX/ PAINTWORKS  
BLOAD PAINTWORKS,T\$B3, A\$1000, L\$1000  
POKE 6621,96  
BSAVE PAINTWORKS,T\$B3, A\$1000, L\$1000**

\*Iigs Softkey for...

**Music Studio**  
Activision

**1** Make a copy of your 3 1/2" original disk. Ignore any errors on block 7.

**2** With a ProDOS block-editor, find byte \$14 in block \$44D. Change it from F0 to 80 and write it back to the disk. If you don't have a block-editor, use a sector-editor to find the sequence 0C 00 C9 01 00 F0. Change the F0 to 80.

\*Iigs Softkey for...

**Shanghai**  
Activision

**1** Make a copy of your 3 1/2" original disk. Ignore any read errors.

**2** With a ProDOS block-editor, find byte \$E1 in block \$243. Change it from F0 to 80 and write it back to the disk.

\*Iigs Softkey for...

**Reader Rabbit**  
The Learning Co.

It appears that the Learning Co. is reading your magazine. A few months ago I sent in a crack for their program *Reader Rabbit*, and now they have changed the copy-protection on it.

**1** RUN COPYA and press [ESC] in answer to the prompt to interrupt the program.

**2** Delete line 70 and tell DOS to ignore errors.

**70  
POKE 47426,24**

**3** Now RUN the program again.

**RUN**

**4** With a sector-editor, find the sequence BC 8C C0. I found it on track \$13,

sector \$D, byte \$53. Change the BC to 60 and write it back to the disk.

Softkey for...

**Math Rabbit**  
Learning Co.

This method also works on *Math Rabbit*, another Learning Co. program. I found the same sequence on track \$A, sector \$0B, byte \$44. Change the BC to 60 and write it back to the disk.

† Softkeys for...

**Baseball Database**  
Jacobsen Software Designs

**Mastery Arithmetic Games**  
Mastery Development

I almost forgot. Here is a crack for two other programs.

The *Baseball Database* from Jacobsen Software Designs and *Mastery Arithmetic Games* from Mastery Development can both be cracked with the *Super IOB* softkey for **Side B** of *Sword of Kadash* from **COMPUTIST #27**.

No other mods are necessary, just run *Super IOB*, choose side **B** when asked, and copy the disk.

**Guy Kendall**

Softkey for...

**James Bond 007**  
A View To A Kill  
Mindscape

This program is a text adventure and a pretty lame one so far. I haven't gotten too much into the game because the program is sooooo slow. I keep thinking it's locking up because it will give me a prompt and then sit there without accessing the drive.

The program is written in UCSD Pascal. I looked at it with Apple Pascal and listed the files but I don't know enough about Apple Pascal to change anything.

☹ Maybe someone could find a way to put a faster Pascal on it.

First I got out *Bag of Tricks* and used *TRAX* to examine the prologues and epilogues. I found that the address and data epilogues had been changed from DE AA to FF FF. So I knew that I needed to tell DOS to ignore this.



I decided to have DOS ignore all epilogue and checksum errors by changing \$B942 from 38 to 18 and running *COPYA*. It copied fine but when I booted the copy it just kept re-booting. This told me that there must be a nibble-count of some sort.

I got out my trusty *Copy II Plus* sector-editor and started looking around. I searched for a reference to the re-boot routine (\$C600) and sure enough I found a 4C00C6 (JMP \$C600) at track \$00, sector \$0C, byte \$94. I replaced this with EA EA 60. This is NOP NOP RTS where NOP is No OPeration and RTS is ReTurn from Subroutine. This was all it took to get a working copy. Have fun.

**1** Tell DOS to ignore all epilogue and checksum errors.

**POKE 47426,24**

**2** Use *COPYA* to copy the entire disk.

**RUN COPYA**

**3** Make the following sector-edits.

Trk	Sct	Bytes	From	To
\$00	\$0C	\$94	4C 00 C6	EA EA 60

That's all it takes.

## The Dabblers

Softkey for...

### Might And Magic

Activision

I recently deprotected *Might and Magic*, a role-playing computer game in the genre of *Wizardry*. Deprotecting it simply consists of changing one byte on the program disk.

#### Requirements

- Four blank disk sides
- A Fast copy program
- Original *Might and Magic*

**1** Copy all sides using a fast copy program, such as *Disk Muncher*, *Copy II Plus* copy disk, etc. Ignore any error on track \$0D (decimal 13).

**2** Search copy of *Disk A* (Program disk) for D0FB E6 01. On mine it was located on track \$0, sector \$F, byte \$87.

**3** Replace D0 (branch command) with a 60 (return from subroutine). Write this sector back to disk.

**4** Put away your originals and breathe a sigh of relief.

## Magnus Hedlund

First, I'd like to congratulate you on a great magazine. I've really enjoyed it and just renewed my subscription.

Here are some hints for *The Coveted Mirror* and *Ultima IV*.

Playing Tips for...

### The Coveted Mirror

- "LOOK" at things in your prison cell to help get out.
- "TALK" to everyone you meet.
- Map out the twisty passages entirely.
- Read the Magician's book for more hints.
- And, if anyone knows how to become invisible, please tell me.

Playing Tips for...

### Ultima IV

Broderbund

- Frequently visit Lord British and ask him to "HELP" you.
- To heal all of your party, ask Lord British "HEALTH" and then say "NO".
- Map all of the moon gates (they each last 3 cycles).
- To have everyone possible join you, You must be 8th level.
- A secret door has a white dot in the wall right in the center.

## Jon Bernhardt

I have COMPUTIST #1-4, and #9 to your most recent and I think they are great. I have a lot of original programs backed up thanks to your magazine. Although I don't know assembly Language very well I was able to back up a program on my own.

Softkey for...

### Award Maker

Baudville

Thanks to J.J. Gifford (COMPUTIST #37)

and William Forsyth (COMPUTIST #32), I was able to look in the right areas and fix my original copy of *Award Maker* by Baudville.

I found the copy protection on track \$0 sector \$5. By using the sector editor on *Copy II Plus* I changed the two jump statements at byte \$76 and \$8A.

Trk	Sct	Bytes	From	To
\$00	\$05	\$76-	20 91 09	EA EA EA
		\$8A-	20 91 09	EA EA EA
		\$5C	?	60

I added a return at \$5C because there was a return at \$5B and this avoided the copy-protection that followed \$5C. I could not find the jump to \$5C. The jump at \$5C gets you past the "NOT AN ORIGINAL DISK" message and the removal of the two jumps allows you to get past the junk border that they use.

Keep up the good work and I would like to see more articles that explain things.

## Brian Symonds

Softkey for...

### Principal's Assistant

Learning Well

This poster-, certificate-, and award-making program gives darker printouts than the *Certificate Maker* (at least on my printer). It is easy enough for a child to use, and of course is copy protected, and comes without a backup. It is difficult to copy with a copy program.

The protection scheme involves using changed address-field epilog bytes on all tracks, and changing the first byte of the address-field prolog from D5 to D4 on odd-numbered tracks. Make your *COPYA* backup as follows:

- 1** RUN COPYA
- 2** After the disk drive stops, halt the program.  
 C
- 3** Remove line 70 so *COPYA* won't reload *COPY.OBJ*  
70
- 4** Enter the monitor:  
**CALL-151**
- 5** Make the following DOS changes.
- 5a** Patch DOS to ignore the first byte of the address header:

**B954:29 00**



**5b** Patch DOS to ignore the address-field epilog bytes

**B988:18 60**

**6** Re-enter BASIC and start up the altered *COPYA*:

**☐**

**RUN**

**7** Be sure to copy both the **Master** side of the disk and the **Art** side of the disk.

I always make a couple of copies of the protected disk with different boots of my disk with *COPYA* on it, then use the 16-sector utility compare program on *Locksmith* to compare the copied disks. I recopy any sectors that do not exactly compare. Occasionally it seems that the DOS patches lead to erroneous reads of the protected disk.

**John Einstein**

I was bored one afternoon so I decided to attempt deprotecting *Conflict in Vietnam* by Micro Prose.

Softkey for...

## *Conflict In Vietnam*

Micro Prose

### Requirements:

- Fast copier
- Sector-editor (with file follower)
- One blank disk

Deprotecting *Conflict* did not take long (only an hour and a half) because I made sure I did a little research into Microproses' protection schemes first.

Because the boot looked similar to that of *Silent Service*, I used that softkey as a guide.

The first thing I did with the disk (as I do with all of my protected programs) is try to back it up and get it into DOS 3.3 format. As it already is in DOS 3.3 format all that was necessary was to use a simple data mover (i.e. *COPYA*) to copy the disk (both sides).

Next, I simply booted the disk to see where it crashed. It crashed just after loading DOS and the boot up program (which was called //, more on that later).

Because the disk was in DOS 3.3 format with a standard file structure (catalog track at \$11). I simply found the bootup program (whose file name is located in the first three tracks of the disk, where you will find DOS) and using the

file follower with my disk editor (*SNOOPY*) tracked down the file and the located valid program data.

One small problem I could have had if I did not use the file follower became apparent when I cataloged the disk. There were two files called // one of them was located at track \$FF (as my sector editor told me). I attacked the other one.

**BLOADING** didn't work. The phony file gets in the way. I found that the program I needed to work with was located at track \$0F, sector \$02 (the file follower goes to the first sector of the file which was sector \$03, go backwards like DOS does through the disk and you should get program data.

Now, I knew I had found the program because I saw at the end of the program Microproses welcome message "HARDWARE FAILURE!". Bingo!

After that I checked for the usual condition branches that sends the program into never never land and displays the welcome message "HARDWARE FAILURE!".

I found that the author of this protection scheme used three BCSs' (Branch if Carry Set) at bytes \$67, \$6F, and \$97 to keep the owner of the disk honest (and in the dark). If, after checking the disk three times, the protection scheme finds no problem with what it sees on the disk, the program will follow through and execute the game.

I simply NOPed each of the BCS instructions.

**1** Copy the disk (using *COPYA*, *Locksmith's FASTCOPY*, etc.). Both sides!

**2** Get out your sector-editor and NOP the BCSs.

Trk	Sct	Bytes	From	To
\$0F	\$02	\$67-	?	EA EA
		\$6F-	?	EA EA
		\$97-	?	EA EA

That's all.

**Bill Beach**

Softkey for...

## *The American Challenge*

Mindscape

After receiving *COMPUTIST #40* a couple of days ago, and seeing another softkey listed in the input section for *The American Challenge* I decided to try again. The one in *COMPUTIST #33* did not quite do the job. Lo and behold, *COMPUTIST #40* did not help much either. Both programs did allow the beginning to load

in but did not take care of the next protection routine which is used just before starting to play.

The softkey in *COMPUTIST #40* did give some important information which is used in this softkey along with the removal of a JSR to \$02F6 which seems to do a nibble count somewhere on the disk.

The controller listed here worked for me.

### Controller

```

1000 REM THE AMERICAN CHALLENGE
1010 TK = 0 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1020 RESTORE : GOSUB 170 : GOSUB 490 : GOSUB
      610
1030 T1 = TK : TK = PEEK (TRK) - 1 : RESTORE :
      GOSUB 310 : TK = T1
1040 GOSUB 230 : GOSUB 490 : GOSUB 610 : IF
      PEEK (TRK) = LT THEN 1060
1050 TK = PEEK (TRK) : ST = PEEK (SCT) : GOTO
      1020
1060 HOME : PRINT "COPY DONE" : END
5000 DATA 255 , 255 , 255 , 255
5010 DATA 21*CHANGES
5020 DATA 0 , 11 , 36 , 234 , 0 , 11 , 37 , 234
5030 DATA 0 , 11 , 41 , 234 , 0 , 11 , 42 , 234
5040 DATA 0 , 11 , 47 , 234 , 0 , 11 , 48 , 234
5050 DATA 0 , 11 , 57 , 234 , 0 , 11 , 58 , 234
5060 DATA 0 , 11 , 71 , 234 , 0 , 11 , 72 , 234
5070 DATA 0 , 11 , 84 , 234 , 0 , 11 , 85 , 234
5080 DATA 0 , 11 , 93 , 234 , 0 , 11 , 94 , 234
5090 DATA 0 , 11 , 108 , 234 , 0 , 11 , 109 , 234
5100 DATA 0 , 11 , 123 , 234 , 0 , 11 , 124 , 234
5110 DATA 7 , 14 , 116 , 234 , 7 , 14 , 117 , 234 , 7
      , 14 , 118 , 234
  
```

### Controller Checksums

1000	- \$356B	5030	- \$7C56
1010	- \$2544	5040	- \$73AD
1020	- \$73B0	5050	- \$4766
1030	- \$7B6E	5060	- \$DEC9
1040	- \$42F4	5070	- \$B62F
1050	- \$A513	5080	- \$5790
1060	- \$148C	5090	- \$C70A
5000	- \$8C24	5100	- \$4FBA
5010	- \$571B	5110	- \$EA4B
5020	- \$76BD	0	- \$0000

It is possible that the call to the nibble-count routine may be located somewhere else on the other disk. To be sure you can always change line 5010 to **DATA 18 CHANGES**, delete line 5110, and use a disk-search utility to look for 4C F6 02 and change it to EA EA EA.

I hope that this will be the answer for those of you that had difficulty getting this program to work.

Thanks for a great magazine. I read it from cover to cover and then use every bit that I can, then I list it all in the data base.

I recently ran across a protection method



which changes the normal locations of volume, track, sector information on the address header to be track, sector, volume.

Ⓞ Is there any way of modifying Super IOB so it can handle this type of protection?

So far I have not been able to get to the RWTS through boot-tracing because the first sector does some EORing that I can't get past yet.



Anyone have any suggestions on how to overcome this problem?.....RDEXed

**Jack R. Nissel**

Softkey for...

## Arcade Album #1

Penguin

To deprotect *Arcade Album #1*, by Penguin do the following.

**1** Install the controller shown below into *Super IOB*.

**2** Make the following sector-edit to your copy.

Trk	Sct	Bytes	From	To
\$00	\$03	\$42	38	18

### Controller

```

1000 REM ARCADE ALBUM #1
1010 TK = 0 : LT = 4 : ST = 15 : LS = 15 : CD = WR : FAST = 1
1020 GOSUB 490 : T1 = TK : LT = TK + 1 : RESTORE : GOSUB 170 : GOSUB 2000
1025 GOSUB 610 : IF PEEK (BUF) < MD AND LT < 35 THEN LT = LT + 1 : TK = TK + 1 : GOSUB 2000 : GOTO 1025
1030 GOSUB 230 : TK = T1 : LT = 35 : 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
2000 POKE 47445 , 212 + (TK / 2 = INT (TK / 2)) : RETURN
5000 DATA 218 , 170 , 218 , 170
  
```

### Controller Checksums

1000	-	\$356B	1040	-	\$E09C
1010	-	\$E254	1050	-	\$38B7
1020	-	\$6207	2000	-	\$468A
1025	-	\$73EA	5000	-	\$FA96
1030	-	\$2E8D			

† Softkey for...

## Expedition Amazon

Penguin

### Requirements:

- Controller for *Sword of Kadash* in COMPUTIST #27

**1** Use the *Sword of Kadash* controller from COMPUTIST #27. When prompted to enter (B)ack, use the controller for the back side of *Sword of Kadash*. Use this for both sides of *Expedition Amazon*.

**2** Make the following sector edits to side 1.

Trk	Sct	Bytes	From	To
\$00	\$03	\$42	38	18
		\$35	DA	DE
		\$91	DA	DE
		\$54-	4A 49 6A	C9 D5 D0
		\$57-	D0 EF	F0 EA

...and the A.P.T for...

## Expedition Amazon

Penguin

### Alter The Prices for Equipment

The trading post equipment prices are shown in text on track \$0B, sector \$02, bytes \$83 to FF and on track \$1A, sector \$05, bytes \$00-13.

The values are stored as ASCII with the hi-bit clear. IE. \$30 is 0, \$31 is 1, ..., \$39 is 9. To change the cost of the equipment simply change the value shown to the one you're willing to pay.

The price of the LOREN is shown in text as 990000 and in hex as 39 39 30 30 30. To change this to 1 in the text, change the hex to 31 20 20 20 20.

**Jerome Thelia**

Softkey for...

## Might and Magic

New World Computing/Activision

### Requirements

- Sector-editor
- copy program that will ignore errors and skip tracks.

*Might and Magic* is a multi-player fantasy role-playing simulation in the dungeon delving tradition of *Bard's Tale* and *Wizardry*. After forming a party of six able men and women, it is your task to take them on a lengthy exploration of a variety of terrain and perils. Your search will be for the Inner Sanctum and the key to its secret.

Since only the boot side is protected, sides B, C, and D can be duplicated using *COPYA* or similar.

The protection is similar to that of some Accolade software. Except for track \$0D, the disk is of standard format.

Soon after the title screen is displayed, the program code performs a JSR to the protection check. If unsuccessful, it reboots. This JSR can be cancelled buy a NOP statement, liberating the disk from the evil shackles of copy-protection.

The irregular format of track \$0D will produce an error and interrupt the copy procedure if using *COPYA*, so you must use a copy program that can skip the track or ignore the error.

When the copy is made, you can implement the required changes in code with a sector-editor.

**1** Copy Side A ignoring the error on track \$0D.

**2** Make the following sector-edits to bypass the protection check.

Trk	Sct	Bytes	From	To
\$00	\$0A	\$AB-	20 00 8E	EA EA EA

That's it.

**Matt Evans**

*Spindizzy* is a fun game to play, but it can be repetitious if you aren't very good. Of course, game disks can get thrashed about and so I decided to try and softkey this one.

Softkey for...

## Spindizzy

Activision

### Requirements:

- FID or some other file copy program
- A blank disk

First, let me say that this is a classic example of putting too much strength into a problem without first trying the easiest method.



The first thing I did was to try a nibble-copier. The copier didn't copy the program very well. I tried another nibble-copier with no luck either. What happened with the copy I did make, was that after a few seconds of boot, the drive would make its infamous noise stating that it didn't like what was on the current track. Hmmmm...

After viewing the errors the nibble-copiers recorded, I thought the program was doing some sort of nibble-counting. I then booted the disk again and interrupted the boot with my *Wildcard* to see if I could find any interesting code that might lead me somewhere. I couldn't find much.

Well, after a few more similar efforts, I tried a track-dump program found in the *Beneath Apple DOS* book. Investigating the track-dump code from track 0, all the correct sector headers were there. You know the familiar D5 AA 96 etc. I wondered if the disk was not protected at all, and catalogable.

I booted up my DOS 3.3 system master to get DOS 3.3 into the machine. Then, I put the *Spindizzy* disk into the drive and typed **CATALOG**. Guess what? The disk is CATALOG-able! So far so good, but I still need to be able to load the programs from BASIC. It turned out that all the programs can be loaded, re-saved, locked, renamed and everything; merely the program wouldn't boot. Hmmmm...

From experience, this tells me that one of the programs on the disk exist merely as a 'disk-checker' program or nibble-counter program that verifies the original disk. Generally this is done either by a nibble-count routine or a trick checksum routine. All I had to do was disable that routine or get rid of the entire program and that should do the trick.

## LOADER?

On the *Spindizzy* disk is a file called **LOADER**. This sounds too easy. I **BLOADED** **LOADER** and checked the **BLOAD** address by entering the monitor and checking locations \$AA72 and \$AA73. The program started at the usual address of \$800. If the program does a nibble-count or checksum, usually the code is easy to find, then it is time to dissect the code. The part that makes it easy to find the special disk code, is to look for a **LDA \$C08C,X**. That is the indication that the program is reading something from the disk on its own and not through DOS.

I begin to examine the code at \$800 to find something special. The first few lines merely set up the reset key so that you can't bomb the program. Then, the address of the data table that **LOADER** uses is put into locations \$10 and \$11 (\$08FA). Then, the series of **LDA**'s in the next few lines load data from the data table and perform a **JSR \$FDED** which means they are printing something.

What? I went to the data table at \$8FA to find

out. Located here are the names of ALL the other files on the disk.

**LOADER** loads in all the files into memory, then, once a 0 is found in the data table (tested at \$840), the program continues at \$85A.

This, so far, is pretty much straight forward. You can see that the code is just looping and there is nothing special (**LDA \$C08C,X**) in the code.

**BUT...** when the code continues, all the programs have been loaded in and a **JSR \$C00** occurs. This is the next stage in the boot.

Now, I **BLOAD SHELL** at **A\$C00**. I found out that it loads here by examining the data table which specifies all the addresses of all the programs.

I begin looking through the code at **\$C00**. Again, immediately the reset is played with again so the program continues execution instead of rebooting. The program can do this because it manipulates reset by copying the monitor ROM into the RAM card and then modifying those parts of the ROM code that weren't meant to be changed.

Anyway, a tricky little bit of code comes next that, after deciphering, sets up an IOB for DOS 3.3, for a track \$4 'seek' (not read or write, just seek). Then, as expected, a **JSR \$C41** occurs which results in a **JSR \$B7B5** which is a call to DOS's **RWTS** routine that controls disk access.

So, the special routine is beginning to unfold, though we haven't found the core.

The program moves the drive head to track \$4 but nothing happens...yet.

After the **JSR \$C41** call, a little playing takes place to clean up the IOB. Then, the kicker... A **JSR \$C51** is seen.

I began looking at the code starting at **\$C51**. The first instruction is **LDA \$C089,X**, which isn't EXACTLY what we are looking for, but it is extremely close.

Anyway, that command turns on the disk drive so it starts to spin, whether or not it was spinning in the first place.

Then, the special code begins. Lots of playing around and a few **LDA \$C08C,X** (or in their case **LDY \$C08C,X**; this performs the same feat though, read a byte off the disk). My main concern is the comparisons done after the **LDY \$C08C,X**. The things that get compared are not familiar bytes that should be on a disk. Also, after the five **LDY \$C08C,X**'s, there is a big giveaway. At **\$C96** here is the code:

```
0C96- 49 AA   EOR #5AA
0C98- 8D BD 0C STA $0CBD
```

## Checksum or Nibble-Count

This is the giveaway that the code is a checksum and not a nibble-count. What I assume now, is that somewhere else in the program, **\$0CBD** is checked to make sure it is equal to a certain number. If it is, the program

continues; if it isn't, the program aborts or does whatever it wants to.

There are two ways to find out what that number should be.

You can continue to boot-trace the code until you find the comparison, and it can be just a few more lines to a long ways away, or you can stop the program here and find out what is in **\$0CBD**.

This is where I think the copy-protectors fouled up. I chose option 2.

Since the program ties up the reset and I have a *Wildcard*, the easiest way for me to bomb into the program is to use my *Wildcard*. So, during the boot, after I saw the drive head move to track \$4 (I had the drive cover off), I pressed the *Wildcard* button and jumped into the monitor. There was a \$55 in location **\$0CBD**. So, what I tried next was the solution. The critical routine starts at **\$0C51**, so I decided to load the accumulator (A register) with \$55 and continue at the **STA \$0CBD** at **\$C98** so the program would store it itself and continue with the rest of the program as if nothing happened.

So, I booted DOS, changed the program, saved it back to disk and everything worked!

## Step-by-step

- 1** Boot DOS 3.3.
  - 2** Put in a blank disk and write a quick hello program.
- NEW**  
**10 PRINT CHR\$(4);"BRUN LOADER"**  
**INIT HELLO**
- 3** Use *FID* or some other file copy program to copy all of the files from the *Spindizzy* disk to the new initialized disk.
  - 4** Put in the new disk and load in the program called *SHELL*.  
**BLOAD SHELL, A\$C00**
  - 5** Enter the monitor and change some code.  
**CALL -151**  
**C51:A9 55 4C 98 0C**
  - 6** Save the changed file back to disk.  
**B\$AVE SHELL, A\$C00, L\$2F0**

Have fun with *Spindizzy*!

One thing to note. Since the disk access routine is skipped, the drive will stop earlier than normal, meaning you won't be able to start the game immediately after the drive stops like before. You will have to wait about one or two seconds. This does not mean that this softkey makes the program longer to boot, it means that *Spindizzy* left the drive running for no reason until the program was ready to run.



**Darryl Higa**

### Softkey for...

## Computer Prep. for SAT v3.1E

HBJ Inc

A previous softkey in COMPUTIST #39 did not work for mine. It was really way off but maybe that was for another version. This softkey is for version 3.1E.

**1** Copy sides A, B, and C with *Locksmith 5.0 FASTCOPY* (or a similar copier) and ignore the errors on tracks \$0, \$21 and \$22 or copy tracks \$3 — \$20.

**2** Use a sector-editor (I used *Copy II Plus*) and move track \$12, sector \$0 to track \$11, sector \$0 on the duplicate disk.

**3** Copy DOS (preferably a fast DOS) to the disk.

**4** For Side D, first initialize a disk.

**5** Now you must capture their RWTS. Put side D in the drive and:

CALL -151

9600<C600.C6FFM

96F8:A9 00 8D 4A 08 4C 01 08

9600G

6000<B800.BFFFM

**6** Remove side D and insert the disk initialized in step 4 and boot it.

6 P

**7** Save the RWTS:

BSAVE RWTS.SAT, A\$6000, L\$800

**8** Use this swap controller to copy tracks \$3 — \$22.

### Swap Controller

```
1000 REM COMPUTER PREP FOR THE SAT V3.1E
1010 TK = 3 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1020 GOSUB 360 : GOSUB 490 : GOSUB 610
1030 GOSUB 360 : GOSUB 490 : GOSUB 610 : IF
      PEEK (TRK) = LT THEN 1050
1040 TK = PEEK (TRK) : ST = PEEK (SCT) : GOTO
      1020
1050 HOME : PRINT "COPY'DONE" : END
10010 PRINT CHR$(4) ; "BLOAD'RWTS.SAT.A$1900"
```

### Controller Checksums

1000	-	\$356B	1040	-	\$0D4F
1010	-	\$2445	1050	-	\$C7BE
1020	-	\$C908	10010	-	\$4EBC
1030	-	\$FB9C			

Maybe there is an easier way to do this, but I recently started cracking disks and this method works well for me. In a couple of months, I'll probably laugh at this softkey.

### Playing Tips for...

## World Karate Championship

Epyx

To get the maximum amount of points (2000 points) for breaking the bricks, move the joystick up and down when there is a bubble by the judge's mouth. Keep on moving the joystick up and down quickly while he says "Get ready...Go". By pressing **[SHIFT]** and the **[+]** key, the background scenery advances.

To see something s-t-r-a-n-g-e, press **[B]** when you are just a white belt.

### Softkey for...

## Hardball

Accolade

The softkey for *Hardball* by Accolade in COMPUTIST #33 did not work on my version. While I was snooping around, I found a JuMP to SC600. I tried NOPing it and it worked!

**1** Boot DOS 3.3 and enter monitor.

**2** Tell DOS to ignore errors:

B942:18

**3** Run *COPYA* and copy the disk.

**4** Make these sector-edits:

Trk	Sct	Bytes	From	To
\$00	\$00	\$71-	4C 00 C6	EA EA EA

**5** Write sector back to disk.

### Softkey for...

## World Karate Championship

Epyx

Just recently, I got *World Karate Championship* by Epyx from a friend. I looked at back issues of COMPUTIST hoping to find some kind of softkey that would work.

In other softkeys, NOPing out the JuMP to \$C600 would do the trick so I tried it. It didn't work.

Then I came across Contach's softkey for *Koronis Rift*, also by Epyx, in COMPUTIST #34 and noticed how similar the protection scheme was to *World Karate Championship*.

I boot-traced my disk and found the nibble-count at location \$900. I jotted down the first couple of lines and searched through my disk.

I found it on track \$00, sector \$0E.

By changing all the branches to \$97F instead of \$99B, the disk was copyable.

**1** Boot DOS 3.3 and enter monitor.

**2** Tell DOS to ignore errors:

B942:18

**3** Run *COPYA* and copy the disk.

**4** Make these sector-edits:

Trk	Sct	Bytes	From	To
\$00	\$0E	\$24	76	5A
		\$29	71	55
		\$39	61	45
		\$47	53	37
		\$54	46	2A
		\$5D	3D	21
		\$6C	2E	12
		\$7B	1F	03

**5** Write sector back to disk.

### Help wanted for...

## Championship Wrestling

Ⓢ I also got *Championship Wrestling* from Epyx but I can't seem to bypass its protection. It basically has the same kind of protection scheme (nibble count) as the others but it does not have any JuMP to SC600. I think the nibble count routine is at location \$BB00 or track \$0, Sector \$5 on the disk but I still can't crack it. Can anyone help me?

### More A.P.Ts for...

## Rescue Raiders

Sir-Tech

In my version of *Rescue Raiders* by Sir-Tech, I have to type in **BOBBYPIN** to get into the cheat mode.

The APTs by Russell Meyer and others did not work on my version.

Bytes \$6F, \$74, and \$83 on track \$D, sector \$0 has the amount of missiles, bombs, and rockets respectively.

To get unlimited bombs, you must NOP bytes \$48 and \$49 on track \$0F, sector \$04. NOPing bytes \$DB—DC on track \$0F, sector \$03 changed my bullets to napalm.

Also, on track \$0D, sector \$00, byte \$6A contains the amount of fuel you have. By changing this to FF, you don't have to worry about refueling for a long time. But beware, if you land your chopper on the pad to refuel, you're going to get a big bang.

I've noticed that whenever you do this, the game will do funny things: freeze up or reboot.



In COMPUTIST #16 Clay Harrell had a softkey and he had sector-edits to defeat the level problem. I tried typing it in and so far, I haven't had any weird things happening.

Trk	Sct	Bytes	From	To
\$0E	\$0B	\$65	A2 03 BD A2	18 A9 00 8D
		\$69	60 1D A3 60	AC 60 F0 43

By the way, from level 5 on, you get napalm BOMBS. Does anyone know how to get them from level 1 on?

### Playing Tips for...

#### Random

Magnum Software

In this game, you will be asked to give an incantation spell which will enable you to pass certain levels. Here's what I found on my version.

Level to pass	Incantation spell
Level 1	TOMB
Level 2	DROWN
Level 3	OOZE
Level 4	RISK

After that you're on your own.

Dennis Gaunt

### More Playing Tips for...

#### Beyond Castle Wolfenstein

Muse

**Attn: Michael Cook...** Regarding Michael Cook's questions about *Beyond Castle Wolfenstein*: you can kill a guard in front of a door and still open it. Stand a bit to one side of the body and aim your weapon at it, then tap the space bar. Repeat until the guard is no longer in the way. This is also useful to 'drag' dead guards out of sight of other guards.

The only way to use the dagger is to hit the colon key and then stab the guards from behind while you are following them. Typing the changes your weapon from the gun to the dagger. As you overrun guards with your dagger drawn, they die. One of the many clues the seated guards will give you when you bribe them with money (hit **M**) when they ask for your pass is to "stab only from behind."

The toolbox is used on locked doors by aiming your weapon at the door and hitting **ST**. This may or may not unlock the door; do not let any guards see you doing this or they will sound the alarm.

Keys are used (I think) on the circuit breaker panel in the room with the four guards. After opening the panel type **CK**. Somehow circuit breaker #2 is important (according to a tip from a seated guard). If I recall, you do this when the alarms are sounding and it will turn them off.

Finally, the room next to Hitler's room is no problem. Position yourself in the room below the room with numerous guards to the far right side of the door going up. When you enter keep going straight up; the guard you must answer to is always near the top not far from the desk.

### Help wanted for...

#### Gemstone Healer

Does anyone know how to "heal" the gemstone pieces in *Gemstone Healer*? SSI has responded to my inquiries twice but they're not giving anything away on exactly how this is done. Anyone got a softkey for this one yet?? I can't find the nibble count or whatever.

### a question about Billy Tessio's...

#### Light Simulator

COMPUTIST #42, Page 29

Finally, was the *Light Simulator* in the April, 1987 issue for real or an April Fool's joke? I'm not about to enter all those hex numbers if not. Is the result that 3-D image that was on the cover? Obviously, the accompanying text was an April Fool but what about the hex dump?



I was assured by a COPYZ-able backup copy of Billy Tessio that the hexdump is genuine and a light simulator is the result. Gordo Lachance's amazing update to the softkey in the very same issue contributes a variation on a theme: the sector-edits do something strange to the *Light Simulator*. And if you haven't typed it in, I urge you to do so just for the fun of it.

By the way, if you have any other alterations for *Light Simulator*, please send them in for publication ..... RDEXed

### Softkey for...

#### Moebius

Origin Systems

We had our *Moebius* (Origin Systems) disk go bad which was replaced quickly by the company for \$5.00. The non-defective disk copied nicely with the *Ultima IV* controller and the sector edits in COMPUTIST #32. The original disk wouldn't boot to completion which was probably why the deprotection procedure

didn't work for me. Keep up the excellent work!

### Help wanted for...

#### Pool 1.5

Can anyone help with a 1981 game by Innovative Design Software entitled *Pool 1.5*? This is an excellent pool simulation game which allows the player to adjust all parameters of shooting the cue ball. The protection also seems quite good, especially for 1981.

The jacket says 48K Apple, DOS 3.2 or 3.3. A bit copy won't work. The only thing readable with a sector-editor is track \$0, sector \$0. The remainder of the tracks that are readable require the DOS 3.2 option on the *Copy II Plus* sector editor. Tracks \$16 — \$1D and \$22 are unreadable and a hi-res disk scan shows these to be garbage (I think).

I've tried every DOS 3.2 technique suggested in back issues of COMPUTIST to no avail. I've tried using *COPYA* with various addresses altered also. Specifically, COMPUTIST #45, page 6 says to:

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

When I tried this I dropped into the monitor at \$803. What's wrong?



*COMPUTIST #45 had a typo. The correct run address for MUFFIN is \$803, the last command should be 803G not 801G. It sounds like you have a DOS 3.2 (13 sector) disk. If you decide to use a swap controller with *Super IOB 1.5*, you must change the number of sectors from 16 to 13 to match the 13-sector disk you are copying. .... RDEXed*

Next, I tried capturing the RWTS and used a controller for *Twala's Last Redoubt* and a Hartley software "B" (COMPUTIST #32) with no success. With the latter, *Super IOB* just hung with the words "Reading" track 3, sector C.

### † Softkey for...

#### Mission In Our Solar System

Thunder Mountain

*The Halley Project* by Mindscape is being marketed in this area by Target Discount stores under the name *A Mission In Our Solar System* through Thunder Mountain software for only \$8.00!

The softkey for *The Halley Project* by Jason Rosenwald (COMPUTIST #41) deprotects this very nicely.



## A. E.

### Playing Tips for...

#### Alternate Reality — The City Datasoft

Here are some coordinates to help you master *Alternate Reality* courtesy of Pirates Harbor of Westchester.

#### Guilds

Thieves	35n	44e
Blue Wizards	48n	19e
Light Wizards	5n	3e
Green Wizards	43n	12e
Red Wizards	15n	48e
Dark Wizards	22n	34e
Star Wizards	12n	28e
Wizards of Chaos	60n	51e
Wizards of Law	50n	62e
Guild of Order	50n	58e
Physicians	15n	6e
Assassins	3n	56e

Alpha Omega Healer 21n 5e

Death traps 5n 42e  
7n 48e

## The Hack saw

### Softkeys for...

#### Dino Eggs Micro Lab

#### Crisis Mountain Micro Lab

Micro Lab's protection schemes on *Dino Eggs* and *Crisis Mountain* are minimal and simple to remove.

To convert them to a *COPYA* format, disable the DOS checksum and epilogue check and run *COPYA*. This basic technique will copy many programs.

#### POKE 47426,24 RUN COPYA

This will make a copy of *Dino Eggs* or *Crisis Mountain*. After you have accomplished that, you must modify the program(s) to read the regular DOS data marks, which requires a sector-editor.

Make the following modifications:

Trk	Sct	Bytes	From	To
\$00	\$09	\$35	D5	DE
		\$91	9E	DE
		\$94-	18 60	EA BD
		\$9B	E7	AA

Their modified RWTS still de-nibblizes the data abnormally, so to normalize it, make the following modifications:

Trk	Sct	Bytes	From	To
\$00	\$0C	\$FB-	BF 1A	BC 19
\$00	\$0E	\$38-	4C 00 BB	08 B0 8E

### Softkey for...

#### BC's Quest for Tires

Track \$21 is a nibble-count track and contains no data needed by the game. Otherwise, the game is standard DOS 3.3.

Copy the disk (skipping Track \$21) onto a blank. Or, if you want to modify the original, just use *Bag of Tricks'* INIT utility, and format track \$21 on the original disk.

**WARNING:** Trying this on the original is not safe and it would be better to try it on a bit copy.

Now, the disk can be copied by *COPYA*, but it won't boot because of the nibble-count. We'll just NOP the JSR to the nibble-count by changing the following bytes with a sector-editor.

Trk	Sct	Bytes	From	To
\$06	\$07	\$E8-	20 00 96	EA EA EA

That's it.

## Peter Wilding

It's unbelievable! I can't believe the new *COMPUTIST*. There was so much information that I was in a reading frenzy. I couldn't fathom that amount of material in one place. Keep it up, I prefer seeing a lot of softkeys in mass confusion than the 5 or 6 softkeys in previous issues.

I would like to thank **Dudley Brett** for his insight into MECC programs, it was a big help.

### More Playing Tips for...

#### Beyond Castle Wolfenstein Muse

**Attn: Michael Cook...** And here's something to help you with *Beyond Castle Wolfenstein*.

- To switch between the dagger and the gun, type `:` a colon.
- To use the dagger, switch weapons (make sure nobody sees you), point the dagger, and simply run into the guard you're killing from BEHIND. It's quieter than shooting.
- To use the toolkit, type `[C]T`. This rewires the alarm system. You've just gotta find the circuit breaker. I haven't yet found a use for the keys, but to use them you must type `[C]K`.
- One more thing: To DRAG a body away from a closet or just out of sight, stand next to it, point your gun in its direction, and press the space bar.

Keep Hardware Corner going. Some of the boards which Bobby mentions are really neat (Ramworks III does what Instant On does, but for 200 dollars more). These cards are well-worth the price, and I think that they will all be very useful to those people who are curious and want to learn a little more about their computers.

## William Green

I would like to share with your readers the following softkeys for many of the new Activision releases including *Game Maker*, *Hacker II*, *Little Computer People*, *Great American Road Race*, *Spindizzy*, *Master of the Lamps*, and *Rocky Horror Show*.

### Softkey for...

#### Activision software...

**Note:** This came from the softkey in *COMPUTIST* #30 but the reader did not reveal what the original code looked like.

Please for all of you future writers. Do that! The code moves around not only on different games from the same company but sometimes even on the same game. Quite frustrating.

Search for bytes A9 56 85 FD A9. The entire string starts and ends like this: A9 56 85 FD A9 08 C6 FC D0 04 . . . . 38 2A 25 FC.

This entire string must be bypassed. All \$38 bytes of it.

Starting at the \$A9, put in A9 FF 18 90 3D. That should put in a jump to the location following the 2A 25 FC code.

The following list shows where I found the pattern on my disks.





## Quick Softkeys for...

### Game Maker

Track \$21, sector \$05, bytes \$54-58.

### Hacker II

Track \$16, sector \$09, bytes \$D0-D4.

### Little People

Track \$00, sector \$0A, bytes \$73-77.

### Great Road Race

Track \$00, sector \$0A, bytes \$C2-C6.

### Master of the Lamp

Track \$00, sector \$0B, bytes \$28-2C.

### Spindizzy

### Rocky Horror Show

Track \$1C, sector \$03, bytes \$58-5C.  
That's it.

Elvin Chan

## Help wanted for...

### Under Fire: The Campaign

Avalon Hill

*Under Fire* by Avalon Hill is the best computer tactical wargame that I've found and the only complaint is that it had a very, very slow DOS. The softkeys from COMPUTIST No. 37 proved to be very helpful and I was able to use a faster DOS.

Since then I have also purchased two other expansion disks. The same softkey only works however with the first Extended Capability disk. *The Campaign* (Extended Capability II) disk seems to have a different protection scheme. The softkey transferred all the files correctly but when I tried the disk, the game went haywire. It's my guess that something was hidden on the 23rd track but I can't confirm this nor do anything about it.

☐ I was wondering if **John Howard**, who wrote the softkey for *Under Fire*, or anybody else could help me with this situation.

Bill Rossmann

Please ask the readers who send in softkeys to include the manufacturer. This helps when newer software is released to give a starting point. Here are two softkeys that worked just by using that information:

## † Softkey for...

### Bop 'N' Wrestle

Mindscape

Use *Infiltrator* in COMPUTIST #47, page 25.

## † Softkey for...

### GFL Championship Football

Activision

Use *Labyrinth* COMPUTIST #46, page 4. I found all the data on track \$1B, sector \$00, bytes \$54-95.

Also helpful is putting the name, company, and issue number into an Appleworks data base for quick reference.

Robert Paquette

## Help wanted for...

### Dollars and Sense

I have been running *Dollars and Sense* on a Franklin Ace 10000, using a boot disk copied with *Copy II Plus*, not wanting to use my original disk. It has worked fairly well for two years but has a subtle form of error. It occasionally loses an entire transaction or it loses the text entry but not all of the sums and postings associated with it. These errors are hard to find and correct. When it happened twice in one month I decided to go back to using the original boot disk and a fresh copy of the system disk. It is too early to tell if this has cured the problem.

☐ Do you or your readers know if this can happen due to a faulty boot disk, or is it inherent in *Dollars and Sense* itself?

I see that you have removed *Dollars and*

*Sense* from your 'Most Wanted' list. May I vote for putting it back on. One needs a more reliable way than a bit copier to make a copy if the possible resultant errors are so subtle.

Thomas Beach

## Double Your F-8 ROM Space Without Motherboard surgery

The F8 ROM in my Apple II Plus recently went bad, so I decided to replace it with a double F8 image in a 2732 EPROM, as described in Ray Darrah's article *Double Your ROM Space* in COMPUTIST #19.

There was a difficulty, however, as the 74LS02 chip he used to invert the signal to pin 18 of the EPROM appeared to have no free NOR gates on my motherboard. In addition, I did not like the idea of cutting traces or soldering wires directly to the motherboard or EPROM chip, especially when it seemed avoidable.

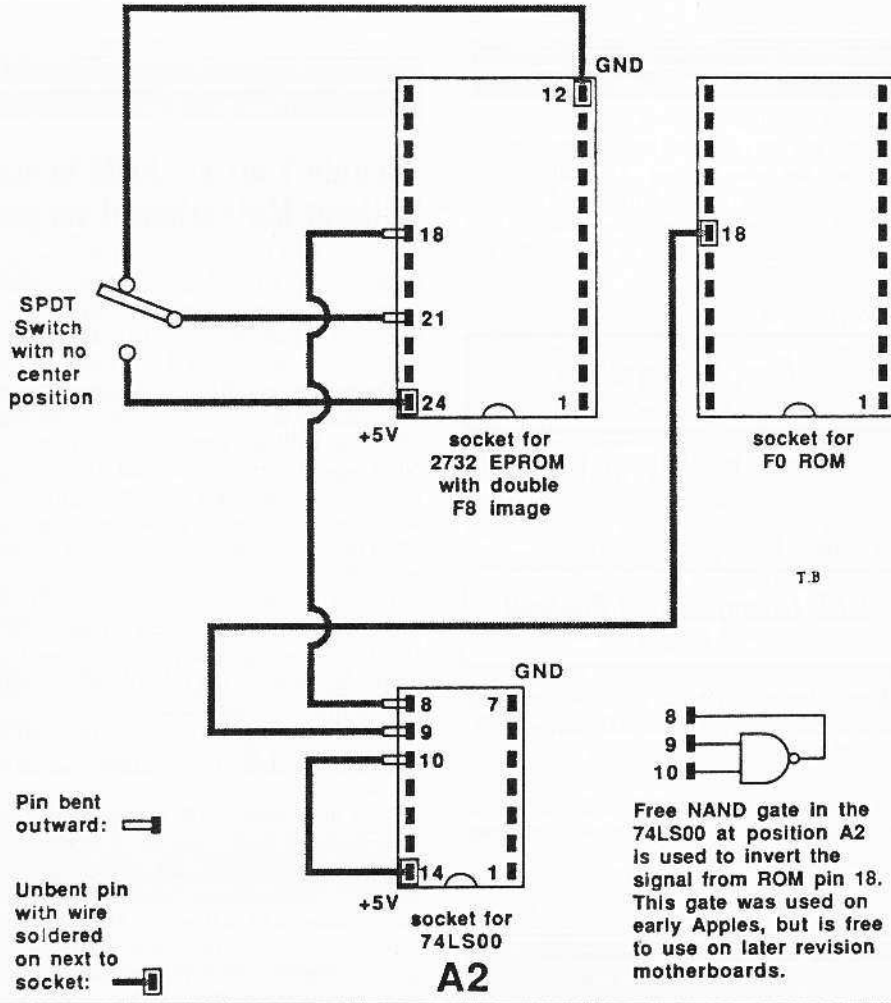
I used the following modification to his procedure. The requirements are the same as listed in his article, except you also need two 24-pin IC sockets, and one 14-pin IC socket. Low profile sockets are preferable.

To invert the signal to pin 18 of the EPROM, I used the 74LS00 at position A2 on my motherboard.

Two dual-input NAND gates in this chip have both of their input lines connected to ground, but the outputs have no connection. On earlier revision motherboards these outputs were connected to 'soft +5V'. The NAND gate I used has its inputs on pins 9 and 10, and its output on pin 8. Check your motherboard (top and bottom) to see that there is no connection to pin 8 to be sure this gate is free to use. By connecting one input to +5V, the gate can be used as an inverter. The other available gate has inputs on pins 12 and 13, and output on pin 11.

Instead of wiring the F8 image select switch directly to the 2732 EPROM as described in the original article, I wired it to a 24 pin IC socket, as shown in the diagram. Carefully bend pin 21 outwards and solder a wire from the pin to the pole of the switch. The wires from the other two connections on the switch are soldered to pin 12 (ground) and pin 24 (+5V) of the EPROM socket. Do not bend these pins! The wires should be soldered as close to the socket as possible, using a minimum amount of solder. Pin 18 is also bent outward and a wire is soldered on. When finished, you must be able to insert the 2732 EPROM into the new socket, and insert this new socket into the old F8 ROM socket on the motherboard, so keep this in mind when bending pins and soldering wires.

Three socket wiring schematic for use of 2732 EPROM with double F8 image. Requires no soldering or trace cutting on the motherboard or EPROM chip.



The 2732 EPROM requires a signal to pin 18 (ROM inhibit) which is inverted as compared to the signal required by the ROMs used in the Apple II.

All of the ROMs have the same signal on pin 18, so I used a second 24-pin IC socket under the F0 ROM to get the signal (do not bend pin 18 on this socket). The signal is inverted by the NAND gate in the 74LS00 at A2.

Access to the NAND gate is gained by preparing a new socket for the 74LS00.

Pins 8, 9, and 10 are carefully bent outward.

Pin 10 is wired to pin 14 (+5V) on this same socket (do not bend pin 14).

Pin 9 receives the signal from the F0 socket, and pin 8 provides the inverted signal to the F8 socket.

Place the 74LS00 in this new socket, and insert the new socket into the old socket at position A2 on the motherboard.

That's it! There is no need to cut the trace

leading to pin 18 of the F8 ROM socket, as described in the original article.

a note on Earl Taylor's...

### Towards a Better F8 ROM COMPUTIST #19, Page 18

If you use the modified F8 ROM described in Earl Taylor's article "Towards a Better F8 ROM" (also in COMPUTIST #19) you should note that there is a discrepancy between the 'new ROM source code' listing, and the hex code listed in the instructions for preparing the ROM.

The Source Code listing has byte \$FF0F as DF, resulting in a new cursor which is an **underline** character, as described in the article.

But the hex code shows the corresponding byte (\$2F0F) as FF, and the new cursor would be a **question mark**, unless you have a modified character generator ROM, in which

case the cursor will display as whatever the FF character is in that ROM. In my modified character ROM, the FF character displays as a **checkerboard block**, similar to the //e cursor.

Lenny Schmidt

More Playing for...

### Beyond Castle Wolfenstein

Muse

You can move the bodies of the SS guards! Simply stand to one side or below them (not touching them), then point your gun toward the guard and press **[SPACE]**. This will move the body to where, if you press **[SPACE]** again, you can search him.

☐ I've played the game many times myself and I always find it a challenge. However, I don't know how to use a tool kit. Can you get inside the room Hitler is in? If so, then how? What should you do if you can get in?

Mr. E. Lau

Softkey for...

### Walt Disney Card & Party Shop

Bantam Electronic Publishing

1 Use Locksmith FASTCOPY to copy side A and side B.

2 Sector-edit side A:

Trk	Sct	Bytes	From	To
\$0A	\$08	\$05-	?	A9 01 8D CF 03 A2 00 4C 01 0A 68

Softkey for...

### Walt Disney Comic Strip Maker

Bantam Electronic Publishing

1 Use Locksmith FASTCOPY to copy side A and side B.

2 Sector edit side A:

Trk	Sct	Bytes	From	To
\$07	\$0E	\$05-	?	A9 01 8D CF 03 A2 00 4C 00 0A 68

This disk is deprotected and COPYA-able.



## Scuzzy Port

\*Iigs Softkey for...

### Deluxe Paint II

Electronic Arts

**1** Make a backup copy of your *Deluxe Paint II* disk, ignoring the bad block errors.

**2** Using a block-editor, scan for the following hex sequence: 08 C2 30 8B 0B DA 5A 4B AB 3B 38. I found it at block \$291, byte \$1B3. Change the 08 to 6B and write the block back to the disk.

**3** Scan for the following hex sequence: A8 F0 02 80 1F. I found it at block \$412, byte \$169. Change the A8 F0 02 to EA F0 00 and write the block back to the disk.

\*Iigs Softkey for...

### Instant Music

**1** Make a backup copy of your *Instant Music* disk, ignoring the bad block errors.

**2** Using a block-editor, scan for the following hex sequence: A8 F0 02 80. I found it at block \$111. Change the A8 F0 02 to EA F0 00 and write the block back to the disk.

**3** Change block \$014, byte \$1DD from 08 to 6B.

*Deluxe Paint II* and *Instant Music* are now unlocked and may even be used from a RAMdisk or hard disk!

## Roy Newberry

Softkey for...

### Print Shop Graphics Library

Holiday Edition

Broderbund

#### Requirements:

- COPYA
- sector-editor
- disk-searcher

The *Print Shop Graphics Library Holiday Edition* is a collection of graphics for just about every holiday of the year.

Unfortunately, the second side of the disk is copy protected and track \$22 is difficult to copy even with a nibble-copy program.

Fortunately for us, Broderbund used the same copy protection system as is used on the *Color Enhanced Print Shop* (see Jason Rosenwald's softkey in *COMPUTIST #46*). The only difference is that the offending byte lies on track \$02, sector \$06, byte \$19. A simple sector edit of this byte from A0 to 60 will solve the problem.

**1** Copy both sides of the original disk with *COPYA* or another whole-disk copy program. (I use the *FASTCOPY* program from *Locksmith 6.0*).

**2** Use your favorite disk-searcher to scan for the byte sequence A0 00 A9 FF and change the first byte to 60.

## Brian A. Troha

\*Iigs Softkey for...

### Thexder

Sierra

#### Requirements:

- Apple Iigs 512K
- ProDOS block-editor with search ability

#### Tools I Used:

- Copy II Plus ProDOS*: was used to make a copy of the 3 1/2" disk.
- SAND.PRODOS*: from Cutting Edge Enterprises was used to search the disk and make the block-edits. This program was invaluable in the procedure.

Sierra has come out with a new game called *Thexder* and it is THE arcade game for the Apple Iigs. I found only two problems with the program, one major and one minor.

First, the program lacks joystick control, which is sad because the game is built for one. You could control all movements with the joystick and have button 0 fire, while button 1 would control the transformation in and out of the 'jet mode.'

Second, the program requires a KEY DISK to play. This means the disk is copy-protected, so if the original fails you are out of luck. I leave it to you, to figure out which is the major problem and which is the minor one. I will explain how you can fix one of these problems.

Like most programs for the Apple Iigs, *Thexder* follows the same type of disk verification, that is: The program comes on a 3 1/2" disk and is protected by checking for a 'BAD BLOCK', block \$634 in this case. The

program will check for the error, and if not found 'knows' it's on a copy. The NOTES sheet says you can upload the program to a hard disk, but you still need a KEY DISK. I was able to eliminate the need for any type of key disk.

To find the protection I loaded the 16-bit system file to a RAMdisk and then searched the disk for ProDOS 16 calls (22 A8 00 E1 [22]), keeping an eye open for block-reads. I came across the following on block \$562:

```

A9 34 06 LDA#$0634 Block to check for
8D D2 10 STA$10D2 Store in parm table
22 A8 00 E1 JSLE100A8 PRODOS 16 MLI
22 00 0022 Command # (read block)
CC 10 00 0010CC Parm table location in mem.
00 00 End of MLI parm table
B0 06 BCS$38 Branch on BAD read
22 0D 10 00 JSL00100D "INSERT MASTER DISK"
80 D0 BRAS$08 Branch ALWAYS to check start
C9 27 00 CMP#$0027
F0 06 BEQ$43
22 0D 10 00 JSL00100D
80 C5 BRAS$08
A9 35 06 LDA#$635 Check next block
8D D2 10 STA$10D2 Store in parm table
22 A8 00 E1 JSLE100A8 Read the block
22 00 0022
CC 10 00 0010CC
00 00
90 06 BCC$5B Branch on GOOD read
22 0D 10 00 JSL00100D
80 AD BRAS$08
22 71 A2 00 JSL00A271
AD CA 10 LDA$10CA
8D CC 10 STA$10CC
A9 34 06 LDA#$0634 Check for bad block
8D D2 10 STA$10D2
22 A8 00 E1 JSLE100A8
22 00 0022
CC 10 00 0010CC
00 00
90 04 BCC$7B Branch on GOOD read
22 3A 10 00 JSL00103A "INSERT GAME PLAY DISK"
A5 0E LDA$0E
A6 10 LDX$10
85 00 STAS$00
86 02 STXS$02
A0 00 04 LDY#$0004
B7 00 LDA[00].Y
29 FF 7F and#$7FFF
97 00 STA[00].Y
18 CLC Set flag for PASS
AB PLB
6B RTL Return to caller

```

This is the heart and soul of the protection, as you can see, the routine checks for the bad block and then makes sure that the next block is readable. If both cases come out 'true', the program starts up and runs the game. Otherwise the program will ask you to "INSERT THE MASTER DISK" and press **RETURN**. If you insert the key disk the program will check it and then ask you to "INSERT THE GAME PLAY DISK", and press



**[RETURN]**. Otherwise the check is run over until it finds the key disk or you press **[ESC]** to cancel the loading. After a lot of looking and some luck, I found the original call is made from the very beginning of the game and is in the form of a JSL 000F42 (relocated later to JSL 030F42). With this in mind the crack becomes very simple:

Block Bytes	From	To
\$55C \$55	22	AF

What the edit does is change the JSL 000F42 to a LDA \$000F42, thus overwriting the call and the game will run. Even when you change the volume name or upload the game to a hard disk you will **never need a key disk**. My first attempt was to edit block \$562 and change the first JSL E100A8 to CLC, PLB, RTL. This method would bypass the check for the bad block (\$634), but if you change the disk name you still need a disk called /THEXDER online. However, with the above edit, even that is no longer needed.

The only tip I will add is: when playing the game, upload the files to a RAMdisk (if possible, ie: if you have the memory) and the program runs without delays when loading in new information. The time spent copying the files to the ram disk will be justified by the increase in the overall speed of the game.

### Softkey for...

## Realm of Impossibility

Electronic Arts

### Requirements:

- Apple II with 64K
- Blank disk
- Sector-editor with disk search
- Fast copier (Copy II plus or equiv.)
- COMPUTIST #49

Realm of Impossibility (ROI) is a newer release from Electronic Arts (EA) with the new low price of \$14.95. In the game you run around collecting spells, keys, and crowns trying to complete all thirteen dungeons. Even with the affordable price, EA still insists on using their infamous protection routine.

Using the same information in the softkey for Arcticfox in COMPUTIST #49, page 16 it's easy to crack ROI. Again the calls to \$569 are required for the game to run right. A list of edits follows:

Trk	Sct	Bytes	From	To
\$01	\$0B	\$4D-	?	18 60 AB
		\$52-	?	18 60 B0
\$01	\$0E	\$4D-	?	18 60 46
		\$52-	?	18 60 4B
\$01	\$0F	\$00-	?	18 60 DD
		\$6F-	?	18 60 DD

### \*Iigs Softkey for...

## Graphicwriter 1.0R & 1.1R

DataPak

### Requirements:

- Apple //gs 512K
- ProDOS block-editor (SAND.PRODOS)
- Copy II Plus

Graphicwriter (GW) by DataPak is a word and graphic processor designed for the Apple Iigs. This program can do color separations for desktop publishing. One unique feature is how graphics are treated. The graphics sections are called regions, and you can move and edit them just like text and place them anywhere on the screen. Although both versions 1.0 and 1.1 don't have the final print routines the program is still quite good. When version 2.0 comes out, DataPak says, they will ship you a copy free of charge if you register your original. Other than that, there is one major problem with GW, it's copy protected.

Most new programs for the Iigs allow you to copy them but you need a 'key disk' (original) to 'unlock' the system. However, with GW the program only bombs into the monitor, this can be both good and bad. When GW crashes into the monitor the super hi-res screen was on, to switch to text mode simply type **[CTRL] [RETURN]**. Now you can see where the program died and the contents of all the registers.

The program disk actually has an unformatted TRACK which could leave 8 to 12 blocks unreadable depending on which track. With version 1.0 blocks \$0C through \$17 are bad, while on version 1.1 it's blocks \$634 through \$63F. The protection routine is called after everything has loaded and just before the program starts. I will use version 1.1 as an example, but the method will apply to version 1.0 also.

The program file starts out like this (the code is in memory block \$03, IE 3/0L to see the following):

```
00:22 8A 0D 00 This value will change when relocated.
...
06:4C 2C 6F
```

The first jump subroutine long (JSL) loads in the program and sets everything up. Then you come to the jump \$6F2C, a look at this new code follows:

```
6F2C: A9 7E 06 LDA #$067E
6F2F: 85 A4 STA SA4
6F31: A2 8A 06 LDX #$068A
6F34: A9 4C 00 LDA #$004C
6F37: 20 75 06 JSR $0675
6F3A: 8D 75 6F STA $6F75
...
6F74: 4C 74 6F JMP $6F74
```

You see a JSR to \$675 and upon return a

value is stored at \$6F75, thus overwriting the endless loop that's there now. So lets check out the code at \$675

```
675: A0 00 00 LDY #$0000
678: 91 A4 STA (A4),Y ; A4+Y=67E,
; A=004C
67A: C8 INY
67B: 8A TXA
67C: 91 A4 STA (A4),Y ; A4+Y=67F,
; A=068A
67E: 60 RTS
67F: 2A ROL
680: 2A ROL
681: 2A ROL
```

Look what the code does, first it stores a 4C 00 at \$67E and then stores a 8A 06 at \$67F. So the RTS is overwritten with JMP \$068A, looking at the next step:

```
68A: A9 00 02 LDA #$0200
68D: A2 85 06 LDX #$0685
690: 20 7B 1C JSR $1C7B
693: AD 87 06 LDA $0687
696: 48 PHA
697: AD 85 06 LDA $0685
69A: 48 PHA
69B: A2 02 20 LDY #$2002
69E: 22 00 00 E1 JSL $E10000
6A2: 20 F5 1F JSR $1FF5
6A5: 8D 83 06 STA $0683
6A8: C0 00 00 CPY #$0000
```

This is where the patch will go

```
6AB: F0 03 BEQ $6B0
6AD: 4C 1A 07 JMP $071A
6B0: E0 43 06 CPX #$0643
```

Check passed, get right value.

```
6FE: A9 BA 70 LDA #$70BA
701: 48 PHA
702: A9 AA D5 LDA #$D5AA
705: 8F 6A 0F E1 STA $E10F6A
709: AD 87 06 LDA $0687
70C: 48 PHA
70D: AD 85 06 LDA $0685
710: 48 PHA
711: A2 02 10 LDX #$1002
714: 22 00 00 E1 JSL $E10000
718: 68 PLA
719: 60 RTS ; A=$70BA
```

Starting with \$6FE you can find the code that gets the right values and returns to the caller. I changed the first conditional branch to a branch always to the code at \$6FE. Your copy of graphicwriter will now boot and run like the original, but without any checks for the bad tracks or blocks. The actual edits for both version are very close:

### For version 1.0R:

Block Bytes	From	To
\$168 \$FB-	F0 03	80 4F



### For version 1.1R:

Block	Bytes	From	To
\$45C	\$F9-	F0 03	80 51

For version 2.0, that is due out soon, a similar trick should work. Find the start of 16-bit system file, then find the first JMP. Follow the code that has similar listings as I have shown. Or use a block-editor with search ability and search the disk for C0 00 00 F0 03 4C, figure out the correct displacement and overwrite the BEQ (F0 03) with a BRA (80 XX). When I get the new version (version 2.0) I will send the information in, so that other readers can make their own backups.

After looking at (and trying to understand) the softkeys for IIGS programs in COMPUTIST #48, I have a few additions I would like to pass on.

### † \*IIGS Softkey Addendum for...

## Hacker II

Activision

First I had a different set of edits for *Hacker II*, they simply ignored the result of the disk check. While *Hacker II* would run using this edit, it was the addition of the edits in COMPUTIST #48 that actually disabled the block read. Just add the following two edits:

Block	Bytes	From	To
\$3D3	\$7D-	F0 12	EA EA
\$451	\$7D-	F0 12	EA EA

### \*IIGS Softkey Addendum for...

## Music Construction Set

Electronic Arts

With *Music Construction Set*, I looked at the code before and after the patch. I found you really only need to change one byte, I chose to NOP the call to the protection because it will fail anyway. You can NOP the codes here because this is NOT a true 16-bit program. MCS boots and runs under ProDOS 8 and is not relocatable like \$B3 type (16-bit program) files.

Block	Bytes	From	To
\$43E	\$CA-	20 00 BF C5 42	EA EA EA EA EA
		49 20 C2 4A 90	EA EA EA EA 80

### † \*IIGS Softkey Addendum for...

## Shanghai

Activision

After checking the code on *Shanghai*, I found

that \$5D09 must hold a value of \$0001. By changing three bytes I stored the correct value at \$5D09 and overwrote the call to the disk check routine.

The edits are shown below with the changes to the disassembly so you can see what's going on.

```
From: A0 00 00 LDY #S0000
      22 74 78 00 JSL $007874
      8D 09 5D STA $5D09
```

```
To:   A0 01 00 LDY #S0001
      AF 74 78 00 LDA $007874
      8C 09 5D STY $5D09
```

Block	Bytes	From	To
\$243	\$1E5	00	01
	\$1E7	22	AF
	\$1EB	8D	8C

### † \*IIGS Softkey Bug for...

## Draw Plus

Activision

Lastly in COMPUTIST #49 I state that to deprotect *Draw* you should change 39 bytes to EA's, well it worked for me. Later I learned this is not the right approach, because IIGS programs are written to be relocatable (with built in relocation tables) some of the EA's will be changed to other values. To fix this I changed all the bytes back to their original values, then changed the first conditional branch to a branch always to the 'ok to start' portion of the code. I am sorry for the mistake, but at least I can show the right patch.

Block	Bytes	From	To
\$516	\$34-	D0 23	80 21

I must thank all those people who sent in the softkeys, with out their information I would not have been able to find the above edits or make backups of those programs. Here are some tips (I found out by trial and lots of errors) for defeating 3 1/2" disk protection:

For subroutine calls, change JSR (20) to LDA (8D) and change JSL (22) to LDA (AF). This allows replacing calls with harmless load accumulator instructions.

```
From: 20 45 6D JSR 6D45
      22 BF 0D 00 JSL 00D0BF
```

```
To:   8D 45 6D LDA $6D45
      AF BF 0D 00 LDA $00D0BF
```

For Conditional branches (BCC, BNE, etc.) you will have to change them to NOP's (EA EA) or BRA xx (80 xx) depending on the situation.

Mathew J. Tomczik

### Help wanted for...

## Ultima III

Origin Systems

☐ I have a question about the first boot strap of the program *Ultima III*. The program does not seem to work on my //c but does work on my friend's Apple //e.

If anyone can help me on this it would be great.

David G. Alexander

### Softkey for...

## Game Maker

Activision

Here is a crack for Gary Kitchen's *Game Maker* which was on the Most Wanted List in COMPUTIST #48.

The protection: a disk check for what appears to be synchronized tracks and is the same routine as in *Great American Cross-Country Road Race* in COMPUTIST #39, page 11. However, the routine is located on track \$21, sector \$5, and starts at byte \$51. The crack is to change bytes \$51—91 from BD 89 C0 A9 56 ... 10 FB to all EA's and bytes \$92—95 from 38 2A 25 FC to EA EA A9 FF.

### Softkey for...

## Microzine #17

Scholastic

The protection scheme on *Microzine #17* is different from that of previous *Microzines*.

The primary protection consists of address trailers changed from DE AA to AB FF. The secondary protection is a binary boot file *HELLO3* which appears to perform a disk check on track \$23, run an AppleSoft file *HELLO* if the check is successful and reboot if the check is unsuccessful.

Copy both sides using *COPYA* with \$B942 changed from \$38 to \$18 (*POKE 47426,24* ignore altered address trailers) then sector-edit side 1.

Trk	Sct	Bytes	From	To
\$0	\$3	\$91	AB	DE
\$0	\$3	\$9B	FF	AA
\$0	\$B	\$42	34	06
\$1	\$7	\$7A	B3	A0

Edits 1 and 2 patch DOS to read the normal address trailers. Edit 3 converts the boot file type from binary to AppleSoft. Edit 4 changes the boot file name from *HELLO3* to *HELLO* by deleting the "3" at the end of the file name.

## Computist in Colorado

Softkey for...

### Robot Odyssey I v2.0

The Learning Company

This robot construction game by the Learning Company is the Best! Not to mention its educational value!

*Robot Odyssey I (Version 2.0)* is protected almost the same as the previous version discussed in COMPUTIST #28, page 11.

Using the information in that issue and a sector editor with a search string function, (*Copy II Plus v7.4*) I found the protection on track \$07, sector \$04, byte \$99.

There are three (3) sides to the new version (A, B, & C), and a demo side that is not protected. Sides A, B, & C had the protection at the same location. You must deprotect all three sides or the soldering iron, in the program, will not work.

Using what has gone before and what I discovered, here are the instructions.

**1** Load in *COPYA*.

#### LOAD COPYA

**2** Add the following lines:

199 POKE 47426,24

249 POKE 47426,56

259 POKE 47426,56

**3** Copy all three sides with this modified *COPYA*.

#### RUN

**4** Use your sector-editor to change the following on all three sides (A, B, & C).

Trk	Sct	Bytes	From	To
\$07	\$05	\$99	18	38

**5** If the string 18 BD 88 is not at byte \$99, then you must use your sector-editor's search function (called *SCAN* in *Copy II Plus*' sector-editor) to look for the string 18 BD 88 and then change the 18 to 38.

You now have a *COPYA*-able and workable *Robot Odyssey v 2.0*.

regarding J. Howard's softkey for...

### Under Fire

COMPUTIST #37, Page 11

⊗ Now I have a question. What are the names of some DOS Utilities that move DOS

into the language card, as required to deprotect *Under Fire* (Avalon Hill), in COMPUTIST #37, page 11, top left?

## Tamaki Murakami

### Some kudo's for The Senior PROM

First of all, I must say that many people haven't realized the importance and the power of *Senior PROM*. This piece of hardware enables you to softkey programs that use sophisticated protections in a few steps. This is a must especially for beginners, who might not even know what RWTS means.

Also, I must ask why there aren't more softkeys that utilize *Senior PROM* in this magazine? *Senior PROM*'s capability of being able to read with modified DOS and write with normal DOS is a help for all novice users as well as for hackers.

Softkey for...

### Amazon

Spinnaker

My version of *Amazon* has protection only on Side 3, the boot side. Using the nibble option of *Senior PROM*, I found that odd tracks are normal DOS 3.3 format. On even-numbered tracks, however, the prolog bytes were D4 AA 96 instead of the normal D5 AA 96.

As explained in the article in COMPUTIST #39, this game's RWTS can read both normal and modified DOS tracks; therefore, normalizing the modified tracks won't cause any trouble at all.

First, copy all odd-tracks with any fast-copier or bit-copier. I suggest one that can skip EVEN-tracks.

Then choose the 'Alter Prolog Bytes' option from *Senior PROM*'s main menu and enter D4 AA 96 D5 AA EB.

Then choose the 'Copy Disk' option to duplicate even tracks. This automatically reads with modified prolog bytes and writes with normal bytes. It's rather a tedious routine to copy only even tracks because *Senior PROM* has no increment tracks option.

This program can be softkeyed if you are able to read with modified DOS and write with normal DOS through assembly language. But for novices, *Senior PROM* is much faster and, most importantly, easier. The documentation that accompanies the hardware is worth the purchase by itself, and all the utilities in ROM are just too convenient.

As Bob Verplank in COMPUTIST #42

pointed out, *Senior PROM* deserves more positive reviews.

### Step-by-step

**1** With any copier, copy all odd tracks.

**2** From *Senior PROM*'s main menu, select 'Alter Prolog Bytes' option and enter:  
**D4 AA 96 D5 AA EB**

**3** Choose 'Copy Disk' option from the main menu and copy even tracks on the Side 3 (boot side) of *Amazon*.

**4** Enjoy the softkeyed version of *Amazon*!

## Dick Lavallee

Iigs Softkey for...

### The Printshop (ProDOS 8 v1.4)

Broderbund

#### ■ Requirements:

- ProDOS sector-editor for 3½" disks (*BYTEZAP.PRO* modified as in page 30 COMPUTIST #48, Mark Swanson).
- Disk-to-disk copier for 3½" disks (*Copy II Plus 7.4*, *Diversi-copy 3.2*, etc.).
- Blank 3½" disk.

### Step-by-step

**1** Copy *The Printshop* disk to disk.

**2** Place the original in a safe place.

**3** With a 3½" ProDOS block-editor make the following edit:

Block	Bytes	From	To
\$2F	\$37	48	60

**Note:** If you don't happen to have a 3½" disk-editor but you do have a 5¼" disk-editor, here's another way to complete the softkey.

#### ■ Requirements:

- ProDOS-formatted 5¼" disk.
- Sector-editor for 5¼" disk.

Complete steps #1 and #2 above then continue with step #3 below.

**3** Copy the file *MF* from the duplicate 3½" disk to the 5¼" formatted disk.



**4** Search the 5 1/4" disk for the string  
48 C9 05 00

**5** Change 48 to 60.

**6** Copy the modified MF file from the  
5 1/4" disk to the 3 1/2" duplicate disk.

## S. T. M.

I got the following softkeys from MAUG on CompuServe and thought that other readers might benefit from this information. I have tried these and they worked on my version of the programs.

### \*Hgs Softkey for...

## Printshop GS

Broderbund

Change:

Block	Bytes	From	To
\$2F	\$7E-	20 4F 67	20 57 68

### \*Hgs Softkey for...

## Draw Plus

Activision

Change:

Block	Bytes	From	To
\$516	\$2D-	2B AD E8 0C C9	00 AD E8 0C C9
	\$35-	23 AD 84 00 48	00 AD 84 00 48
	\$56-	02 AB 60 E2 20	00 AB 60 E2 20
	\$A9-	18 FB C2 30 0B	6B FB C2 30 0B

### \*Hgs Softkey for...

## Reader Rabbit

Learning Company

Change:

Block	Bytes	From	To
\$041	\$125	F0	80

### \*Hgs Softkey for...

## Writer's Choice Elite

Activision

Change:

Block	Bytes	From	To
\$523	\$73-	C9 07 00 D0 01	EA EA EA EA EA

## James Slate

I used the *Spindizzy* softkey (COMPUTIST #48) to crack two other Activision programs.

### Softkey for...

## Aliens

Activision

**1** Fast copy or COPYA both sides of *Aliens*.

**2** Use your sector-editor and make the following changes:

Trk	Sct	Bytes	From	To
\$0B	\$0E	\$58-9A ?		a// EAs
		\$9B-	25 FC	A9 FF

### Softkey for...

## Alter Ego

Activision

**1** Fast copy or COPYA all six sides of *Alter Ego*.

**2** Use your sector-editor and make the following changes:

Trk	Sct	Bytes	From	To
\$13	\$05	\$64-A6 ?		a// EA's
		\$A7-	25 FC	A9 FF

If the tracks and sectors don't match or you have another Activision program, try to search the disk for 38 2A 25 FC and if you don't get a match try just 25 FC.

Then comes the fun part, first look for the last 60 (RTS) before the 25 FC, around 60 bytes in the same sector.

Next change all bytes between the 60 and the 25 to EAs. Then change the 25 to A9 and the FC to FF and write the changes back to the disk.

The copy will now work fine and can be fast-copied. Good luck and Happy Cracking.

## J. Slate & Phillip Doherty

### \*Hgs Softkey for...

## Bards Tale Hgs

Electronic Arts

### ■ Requirements:

- 3 1/2" copy program (*Copy II Plus* or other)
- ProDOS sector-editor (We used ZAP on *Bag of Tricks II*)

**1** Copy *Bards Tale Hgs* master.

**2** Boot *Bag of Tricks* and select ZAP.

**3** Open file to edit:

**OPEN / BARDS.TALE/ BURGER**

**4** Read block \$00.

**5** Change byte \$11 from B0 to 90.

**6** Change byte \$26 from 38 to 18.

**7** Write these changes back to the copy.

**8** Now you have an unprotected copy that can be put on a hard disk.

An optional method for those who don't have a sector editor that will read 3 1/2" drives, first make a copy of the *Bards Tale Hgs* master. Next copy the file *BURGER* to a 5 1/4" disk. Then use any ProDOS sector-editor and search for the string 78 98 E9 1E B0, and change the B0 to 90. Then look for the string E9 20 90 01 38 and change the 38 to 18. The second string should be around 20 bytes after the first string in the same block. Finally, write the changes back to the 5 1/4" disk and copy *BURGER* back to the 3 1/2" drive.

## Jack Moravetz

### Softkey for...

## Minnesota Educational Computing Consortium

(MECC) software...

### ■ Requirements:

- Apple //+, //e
- Super IOB v1.5
- Blank disk

Usually using the Swap controller or a controller similar to Paul Giguere's one for *Puzzles and Posters* (COMPUTIST #46) will allow deprotection of MECC software. The protection scheme appears to be an altered address prolog from the standard D5 AA 96 to AA D5 AD and altered data prolog from D5 AA AD to D5 96 AA. The address and data epilogs



I decided to use the DOS from the original because it had data on tracks \$00, \$01, and \$02. I changed my controller to copy from track \$01 to \$22 and used *Diskmuncher 8.0* to copy only track \$00 to the copy since it was in standard format. Using the sector editor in *Copy II Plus*, I searched for D5 and examined the sector to see if the D5 was part of the address and data field that *Oregon Trail's* DOS was looking for.

Since the rest of the copy was now in standard format, track \$00 needed some changes to make the copy boot.

These are the sector-edits that the MECC controller will make:

Trk	Sct	Bytes	From	To
\$00	\$06	\$86	96	AA
		\$8B	AA	AD
\$00	\$07	\$1F	96	AA
		\$2A	AA	AD
		\$83	AA	D5
		\$8D	D5	AA
		\$98	AD	96
\$00	\$0A	\$A8	AA	D5
		\$AD	D5	AA
		\$B2	AD	96

I came across many MECC programs with the same protection scheme, so I decided to include the sector-edits in the controller to save time and trouble.

Here is a list of all the MECC programs that this controller has completely deprotected:

*Arithmetic Critters*

*Counting Critters*

*Clock Works*

*First Letter Fun*

*Fun From A To Z*

*Jenny's Journey*

*MECC Dataquest: Fifty States*

*Number Munchers*

*Paint With Words*

*Path Tactics*

*Phonics Prime Time*

*Quickflash!*

*Skylab*

*Sound Tracks*

*Speedway Math*

*Amazing Reading Machines*

*the Market Place*

*Word Munchers*

*Words At Work*

If the address prolog on a protected MECC disk is AA D5 AD and the data prolog is D5 96 AA, there's a good chance that this controller will deprotect it.

### Controller

```

1000 REM MECC CONTROLLER
1010 TK = 0 : LT = 1 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1020 GOSUB 430 : GOSUB 490 : GOSUB 610 : T1 =
      TK : TK = PEEK (TRK) : GOSUB 310 : TK = T1
1030 GOSUB 430 : GOSUB 490 : GOSUB 610 : IF
      PEEK (TRK) = LT THEN 1050
1040 TK = PEEK (TRK) : ST = PEEK (SCT) : GOTO
      1020
1050 TK = 1 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1060 RESTORE : GOSUB 190 : GOSUB 210 : GOSUB
      430 : GOSUB 490 : GOSUB 610
1070 GOSUB 230 : GOSUB 430 : GOSUB 490 : GOSUB
      610 : IF PEEK (TRK) = LT THEN 1090
1080 TK = PEEK (TRK) : ST = PEEK (SCT) : GOTO
      1060
1090 HOME : PRINT "COPY*DONE" : END
5000 DATA 170 , 213 , 173 , 213 , 150 , 170
5010 DATA 10*CHANGES
5020 DATA 0 , 6 , 134 , 170
5030 DATA 0 , 6 , 139 , 173
5040 DATA 0 , 7 , 31 , 170
5050 DATA 0 , 7 , 42 , 173
5060 DATA 0 , 7 , 131 , 213
5070 DATA 0 , 7 , 141 , 170

```

```

5080 DATA 0 , 7 , 152 , 150
5090 DATA 0 , 10 , 168 , 213
5100 DATA 0 , 10 , 173 , 170
5110 DATA 0 , 10 , 178 , 150

```

### Controller Checksums

1000	-	\$356B	5010	-	\$B321
1010	-	\$EA41	5020	-	\$CB92
1020	-	\$BBF5	5030	-	\$1250
1030	-	\$8863	5040	-	\$233B
1040	-	\$66DE	5050	-	\$69E8
1050	-	\$8D3B	5060	-	\$AE31
1060	-	\$3332	5070	-	\$0E7B
1070	-	\$80E2	5080	-	\$2145
1080	-	\$1954	5090	-	\$3032
1090	-	\$29D5	5100	-	\$DB96
5000	-	\$713C	5110	-	\$C573

Softkey for...

*Fish Scales*

DLM Software

### Requirements:

- Apple II Plus, //e
- Ability to reset into the monitor
- Two blank disks
- A DOS 3.3 slave disk
- FID from DOS 3.3 System Master

*Fish Scales* is an excellent early learning educational program dealing with numbers and measurement. Usually the Swap controller or the controller for DLM software in COMPUTIST #25 by Greg Prior will allow the deprotection, but they changed the address field epilogs and data field epilogs from the normal DE AA to 97 96. There was some data on track \$2 also. I used Controller Writer from COMPUTIST #16 to write a controller for *Fish Scales*.

They added a nibble count routine to prevent copying the files and running the program. If you try to run your copy, it will only reboot. I found that the file *FISH.OBJ0* was the file that did the nibble count. Not being good at boot code tracing, I found a way to disable the nibble count by other means. It may not be fancy, but it works.

### Step-by-step

**1** INITIALize one blank disk with a normal or fast DOS and put the disk aside for now.

**INIT HELLO**  
**DELETE HELLO**

**2** Install the *Fish Scales* controller into Super IOB and run Super IOB to copy the original onto a blank disk, use the format disk option from within Super IOB.





**3** Using *FID* or file copier, copy all the files except *FISH.OBJO* from the copy made by Super IOB to the disk that was initialized in step 1. We will capture *FISH.OBJO* later.

**4** Boot the original *Fish Scales* disk and reset into the monitor after the title screen is displayed and the disk drive stops spinning.

**5** Insert your DOS 3.3 slave disk and boot it from the monitor.

### C600G

**6** Save the file with the nibble count already performed to your slave disk or the disk that the other files were copied to.

**BSAVE FISH.OBJO,A\$4000,L\$OFFF**

If you saved the file to the slave disk, copy it with *FID* or a file copier to the disk with the remaining files. That's all there is.

### Controller

```
1000 REM FISH SCALES CONTROLLER
1010 TK = 0 : LT = 35 : CD = WR : MB = 151 : ONERR
      GOTO 550
1020 ST = 0 : T1 = TK : GOSUB 490 : RESTORE :
      GOSUB 190 : GOSUB 210 : GOSUB 170
1030 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST
      < 16 THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 : TK = TK + 1 : IF TK < LT THEN 1030
1060 GOSUB 230 : TK = T1 : ST = 0 : GOSUB 490
1070 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST
      < 16 THEN 1070
1080 ST = 0 : TK = TK + 1 : IF BF = 0 AND TK < LT
      THEN 1020
1100 HOME : PRINT "COPY'DONE" : END
5000 DATA 213 , 170 , 150 , 213 , 170 , 173 , 151
      , 150 , 151 , 150
```

### Controller Checksums

1000	- \$356B	1060	- \$5DFB
1010	- \$694C	1070	- \$3FF0
1020	- \$B431	1080	- \$51DA
1030	- \$FF9D	1100	- \$46C5
1040	- \$137E	5000	- \$6276
1050	- \$E3BF		

*Sofitkey for...*

## Electric Crayon ABC's

Polarware/Penguin Software

### Requirements:

- Apple II Plus, //e
- ProDOS and *BASIC.SYSTEM*
- COPYA* or disk copier

*Electric Crayon ABC's* is a double hi-res computer coloring book which can be colored and erased. It also has the ability to print the picture and a calendar.

Since it is primarily a program for children, I saw the immediate need to have a backup. I was disappointed to find that a ProDOS based program such as this was copy protected.

After examining the system program, it appeared that the file *SHAPES* was nothing more than a protection file so I decided to disable it.

### Step-by-step

**1** Copy the original disk with *COPYA* or any disk copier.

**2** Boot the ProDOS Users disk and exit to *BASIC.SYSTEM*

**3** Place the backup disk in the drive and type:

```
PREFIX /A
BLOAD SHAPES,A$9000
CALL -151
9000:18 60
3DOG
BSAVE SHAPES,A$9000,L$00BD
```

*Electric Crayon* is now in a *COPYA*-able format.

Ron Balewski

## Ace-Apple Bi-modal Switching Puts Two Machines in One

I'm the proud owner of a Franklin Ace 1000 series computer. I know it ain't an Apple, but I did save a BUNDLE of money by buying a machine that does almost everything an Apple does. Almost.

My quest for total compatibility came one step closer to reality recently when a friend and I found a set of Apple EPROMs at an area hamfest. Naturally, being true tinkerers, we bought them. Of course they worked perfectly in our Franklins. Our ACEs even said APPLE II when turned on!

But I missed certain aspects of the ACE 1000 ROMs not supported by the Apple ROMs. I could keep swapping ROMs all the time to change from Ace to Apple, but that's too much work. I had to find an easier way.

Fortunately, I did! Now I just flick a switch to either Apple or Ace before powering up the computer, and voila! The desired computer is at my disposal. And the procedure was so simple that, if you have an Ace and a friend

with an Apple, you too can have a "bi-modal" Ace (Apple owners can even install a set of Ace ROMs in their Apple's). Here's how the upgrade works.

Some Franklin Ace's (*see important note*) and many Apple II Plus' (I think) use 2716 EPROMs to store BASIC. 2716s hold 2K 8-bit bytes. You're going to substitute 2732s, which hold 4K bytes. You'll store Apple BASIC in the lower 2k and Ace BASIC in the upper 2K of each EPROM.

Then, to select a BASIC, all you have to do is select either the top half or the bottom half of the new EPROMs. This is quite easy to do. The 2732 has one more address line than the 2716. To select top or bottom, all you have to do is connect this highest-bit address line to either +5v (logical 1) or ground (logical 0). That's it!

Here's what you'll need to do the job:

### Required Components:

- Components:
  - 6 24-pin IC sockets (Radio Shack #276-1989)
  - 6 2732 EPROMs
  - 1 SPDT Switch (such as Radio Shack #275-613)
  - 1 4.7k resistor (Radio Shack #271-030)
  - Wire

### Required Tools/Accessories:

- Apple II or Franklin Ace 1000
- EPROM programming card
- EPROM eraser (maybe)
- Soldering iron
- Misc. hand tools

Now for the procedure. The hardest part will be burning the new EPROMs. You must first assemble a disk containing both Apple and Ace ROM images.

Let's start with your Ace. Put a formatted disk into your disk drive and type the following:

```
BSAVE ACE D0,A$D000,L$800
BSAVE ACE D8,A$D800,L$800
BSAVE ACE E0,A$E000,L$800
BSAVE ACE E8,A$E800,L$800
BSAVE ACE F0,A$F000,L$800
BSAVE ACE F8,A$F800,L$800
```

Now take your disk to your Apple-owning friend's house, put it in his disk drive and type the six statements above, except substitute APPLE for ACE in the file names.

You now have all the files you need to burn your new ROMs. Plug in your EPROM burner and let's go.

First, load in the "D0" ROM images by typing:

```
BLOAD APPLE D0,A$1000
BLOAD ACE D0,A$1800
```

The data for your first EPROM is now loaded into memory from \$1000 to \$1FFF.



Note that Apple is on the bottom and Ace is on top. To write the EPROM, just plug a 2732 into your EPROM burner and write #1000 thru \$1FFF to it.

Burning the other five EPROMs is very similar. Just repeat the above steps, changing each "D0" in the above BLOAD statements to "D8", then "E0", then "E8", then "F0", and finally "F8".

Make sure you mark each EPROM as you program them! They must be installed in the proper order!

Next, you've got to install the new EPROMs in your Ace. Here's the procedure.

## Step-by-step

**1** Remove the cover from your Ace and remove the 6 EPROMs. You should find a box drawn around them and labeled "EPROM AREA".

**2** Insert your new 2732 EPROMs into 24-pin sockets.

**3** Bend pin #21 of each IC socket outward at a right angle. To find this pin, look at the EPROM/socket combination from the top. Position the EPROM so that the notched edge faces downward. Pin #21 is the fourth one from the bottom on the left-hand side.

**4** Plug the EPROM/socket assemblies into the sockets on the Ace. If you sit facing the Ace, the notched edge of the EPROM should be toward you and the bent-out pin should point left. The EPROMs should be inserted in the following order from left to right: F8, F0, E8, E0, D8, D0. The circuit board is labeled, in case you get confused. Just make sure you plug the proper chips into the proper sockets!

**5** Carefully solder jumper wires to the bent pins to connect them all together.

**6** Mount a SPDT switch to your Ace.

**7** Connect a wire from the common terminal of the switch (usually the center terminal) to any of the bent pins (since they're all connected together, they'll all get the same signal regardless of which pin the switch wire is connected to).

**8** Connect either of the other two terminals on the switch to a ground point on the Ace circuit board. I used the lead of part CG3 that faces the keyboard (CG3 is on the far right of the circuit board near the big letter G).

**9** Connect the final switch terminal to +5v through a 4.7k ohm resistor. For +5v, I used the opposite lead of CG3, as described above.

That's it! Now turn on the Ace and see what happens! You'll get either the Apple or the Franklin power-on message.

Try turning off the computer, flipping your new personality switch, and turning on the computer again. You should now get the other power-on message.

You now have the best of both worlds. When you want your Ace to be an Ace, it can. But when you need super-compatibility, just flick the switch and Ace turns Apple!

## Important Note

**Franklin Ace 1000 series computers which have disk controllers built onto the mother board do NOT use six 2716s. Instead, they use three 2732s to save space.**

**You CANNOT use this trick on these Aces. The 2764s that you'd need have more pins than the 2716s and 2732s. Therefore, they wouldn't fit properly in the sockets.**

Softkey for...

### Super Boulder Dash

Electronic Arts

*Super Boulder Dash* from Electronic Arts (EA) is an excellent arcade-type game that I'd recommend to anyone -- especially since it's considered a 'classic' and can be had for less than \$10.00!

Of course, like most games, it's copy-protected. I usually just back up protected software with *EDD 4 Plus*, but EA software always presents problems. Oh, well, I'll just have to deprotect it!

At first I thought I could be lazy and find the deprotect spell in *COMPUTIST*. But my hopes faded when I saw *Super Boulder Dash* on the Most Wanted list. I guess I'm on my own for this one!

Fortunately, *Boulder Dash* seems to use Electronic Arts' standard copy-protection. Deprotecting it was really quite easy. I used exactly the same method as outlined for *Marble Madness* in *COMPUTIST* #47. The only difference is that with *Super Boulder Dash*, both sides must be deprotected. I suppose they protected both sides since each side contains a different game.

To deprotect *Super Boulder Dash*, just do the following to both sides:

**1** Copy the disk using any copy program that can ignore errors or skip tracks (you must skip track \$06). I used *COMPUTIST*'s *Super IOB* with a slightly modified Standard controller.

**2** Make the following patches to the disk with your favorite sector-editor:

Trk	Sct	Bytes	From	To
\$01	\$0C	\$00	4C	60
\$01	\$0F	\$00-02	4C 69 A0	18 60 DD
		\$6F-71	4C 69 A0	18 60 DD

That's it! *Super Boulder Dash* is now deprotected!

⊕ Let me warn you that there may be a problem with the game on the back side of the disk (*Boulder Dash II*). On my deprotected copy, certain levels don't work. The screen just fills with garbage and the computer hangs up. However, the first side has been working fine. I suspect that I just copied a track wrong somewhere on my disk and haven't tracked it down yet.

I'd appreciate hearing from others who deprotect *Super Boulder Dash*. I'm really curious whether the side 2 problem is indeed a bad copy or whether EA did something really strange on that disk!

A. L. Head

Softkey for...

### Word Handler

Advanced Logic systems

### List Handler

Advanced Logic Systems

## Requirements

- Word Handler*
- List Handler*
- COPYA* or other standard copier
- Sector-editor with search capability
- Nibble-editor

The Handler package consists of:

*Word Handler*  
*List Handler*  
*List Utilities*  
*Spell Handler*.

The *Word Handler* occupies two sides of a disk. Side A is for those without an 80-column card, and Side B is for those with one.

*List Handler* and *List Utilities* are designed to accommodate all kinds of lists and to manipulate the lists.

*Spell Handler* is a 90,000-word dictionary used to check a document for misspelled words. These programs operate together as an integrated package.



## The Protection

The *Word Handler* and *List Handler* are protected software. *List Utilities* and *Spell Handler* are unprotected and can be copied using standard copiers such as *COPYA*.

The *Word Handler* and *List Handler* are written in a nearly standard DOS 3.3 format and can be cataloged from a DOS 3.3 environment. In addition, both can be copied without error using *COPYA* or *Locksmith 6.0 Fast Disk Backup*. However, both will hang during the booting process.

A catalog of *The Word Handler* shows thirteen files with the boot-up file being an Applesoft file named *WRDHND*. The *List Handler* has two files with the boot-up file being an Applesoft file named *HELLO*.

An examination of the disks using a nibble editor, such as contained in *Locksmith 6.0* or *Copy II Plus*, shows that the sync nibbles used are AA's instead of the usual FF's. This of itself is of little significance unless it is used in the protection scheme. In the large sync field preceding physical sector \$00 of each track a string of nibbles has been inserted as follows:

Disk Nibbles: AA AA AD AA B3 FE AE FE AE FE  
Framing Bits: 2 2 0 0 0 0 0 0 0 0

Disk Nibbles: AE FE AE AC AC AC AA AA AA AA  
Framing Bits: 0 0 0 0 0 0 2 2 2 2

*Locksmith's Framing Bit Analyzer* was used to determine the number of framing bits.

## The Word Handler

The next task is to determine what is happening during the booting process. As mentioned earlier, the boot-up file is an Applesoft file named *WRDHND*. Examination of this file shows that it provides the first menu where selections are made.

The number 1 selection is "RUN WORD HANDLER." If this selection is made, a text file named *WRD* is executed. Examination of *WRD* shows that it *BLOAD's* *WI*, *WA*, and *WB* in that order and then *BRUN's* *WH*.

An interesting characteristic of using text files to control machine operations is that the BASIC cursor is generated for each line executed. This means that four cursors will be seen on the screen when *WH* begins to run. This clue signals when execution of the code begins. When the copy is booted the four cursors are seen as predicted and then the copy hangs.

*My Trak Star* shows that the read/write head is positioned over track \$17. A nonmaskable interrupt (NMI) at this point using the *Senior PROM* shows the Program Counter pointing to address \$5572; the first return on the stack is \$4702.

Remember that one (1) must be added to this to get the return address. Jumping into the monitor and disassembling the code on pages

\$46, \$47, \$55, and \$56 reveals the protection.

Beginning at \$46FD (file *WH*), the disassembly shows a jump to the subroutine at \$5500 (file *WB*). The return address, \$4703, executes a branch to \$473B if the overflow flag is cleared. If the overflow flag is set the execution falls through to an endless loop at \$4705 through \$4708. Obviously, the overflow flag must be cleared on return from the subroutine to continue program execution at \$473B.

This code can be seen by *BLOADing* *WH* from a DOS 3.3 environment and disassembling it beginning at \$46FD.

A disassembly listing of the interrupted program code beginning at \$5500 reveals the heart of the protection being used. This is self-modifying code. The raw code can be seen by *BLOADing* *WB* and disassembling it beginning at \$5500. This code is alright down through \$5525. The modified code can be seen only after the execution of the file.

Notice that \$550A through \$5518 modifies memory from \$5526 through \$5625 by *EORing* *SAC* with the corresponding memory contents from the unmodified code.

This code is further modified through the jump at \$5526 to \$5671. Memory locations \$5529 through \$5627 are *EORed* with *SB5*.

Finally, the code is modified again through the jump at \$5686. The modified code seeks the hexadecimal string AA AD AA B3 beginning at \$5572. This looks familiar. Then, it reads the next \$07 nibbles and stores them at \$565F through \$5659. It runs a comparison and then seeks four consecutive sync AA's. Finally, the execution returns to the caller (\$4703) at \$5657. All of the code on pages \$55 and \$56 is self-contained and deals with the protection scheme. When the return occurs the overflow flag is either cleared or set.

Anyone can see the codes referenced above by pressing  **RESET** at least twice in rapid fashion immediately after the fourth BASIC cursor appears. This is coincident with the hang. This will place the machine in BASIC. Type the following:

**CALL -151**  
**B7EC** track number being accessed  
**46DFI** disassembly listing, see note.  
**5500L** disassembly listing, see note.

**Note:** Press an additional "L" for another screen of listings.

If the second disassembly listing does not agree with the discussion, interrupt the program at a slightly later time. Execution must be in the vicinity of \$5572 for the deciphered code to be present.

In summary, this protection scheme is contained in a file named *WB*. It involves self-modifying code that does read the signature in the sync field before the address of sector \$00 of track \$17. It also checks the sync nibble

before the signature and the four (4) sync nibbles after the signature to see if they are \$AA's. If all the protection hurdles are cleared, the execution returns to \$4703 with the overflow flag cleared.

## List Handler

The *List Handler* has two files. The *HELLO* file simply *BRUN's* *MAIN3*. There is also a deleted file named *MAIN2* on the disk. Booting a copy of *List Handler* and *NMIing* it just as it hangs shows the read/write head over track \$1E, the Program Counter pointing to \$3172, and the first return on the stack at \$3008. The return address will be \$3009. Examining this code shows it to be identical to that for *Word Handler* loaded at different memory locations. The address of \$3000 corresponds to \$46FD for *The Word Handler*. The address of \$3100 corresponds to \$5500. If the overflow flag is cleared on return the program execution falls through to \$3800.

Again, disassembly listing can be obtained for this program by using the  **RESET** approach discussed previously. The BASIC cursors are not available to provide the reset signal. It must be done, however, just as the program execution hangs.

The protection scheme is identical to that used for *The Word Handler*.

## The Deprotection

Standard copiers will be defeated because they will ignore the inserted nibbles. The tracks will be formatted with the usual sync nibbles of FF with no nibbles inserted in the sync field.

The popular bit copiers will do a reasonable job of copying the altered sync field. Both *Locksmith 6.0* and *Essential Data Duplicator III* will successfully backup the *Handlers* most of the time using default settings. *Copy II Plus* will fail. All will work if the proper parameters are set to find the track start/length realistically, reproduce the inserted nibbles, and write the sync nibble as AA. Of course, the goal here is to remove the protection.

The best way to deprotect *The Word Handler* and *List Handler* is to go to the beginning of the protection code, clear the overflow flag, and return to caller. The calling code could be modified to prevent the jump to the subroutine containing the protection code, but on some future version the jump might be placed somewhere else. Here is the way to do it.

Using a sector editor with search capability such as *Locksmith INSPECTOR/WATSON* or the *Senior PROM Sector Editor*, search each copy of *The Word Handler* and *List Handler* for the hexadecimal string 08 68 8D 61. On my program disks this string is located at track \$16, sector \$0E, byte \$04 for *The Word Handler* and on track \$19, sector \$0D, byte \$04 for *List Handler*.



Make the following sector edits and write them back to disk:

### The Word Handler (both sides)

Trk	Sct	Bytes	From	To
\$16	\$0E	\$04	08 68	B8 60

### List Handler

Trk	Sct	Bytes	From	To
\$19	\$0D	\$04	08 68	B8 60

## The problem with track/sector edits

This deprotects the copies. They are already COPYA-able. However, the trouble with track/sector edits for file based programs is the ease with which the files can be relocated on the disk, thereby changing the track/sector location for the edit. Sometimes all the producer of the protected software does on later versions is relocate the files on the disk.

It is better to load the protection files, edit them in memory, and then save them back to disk. This approach frees the softkey from a particular track/sector edit. I have written a Controller for Super IOB 1.5 to copy the disks and perform the file edits. It is presented at the end of this article. A few comments are appropriate.

Lines 1000 through 1040 are the Fast Controller that copies the disks. Lines 1050-1080 set the ONERR flag, identify which Handler is being copied, and load the appropriate file into memory. Lines 1085 and 1090 determine the load address and the length of the file loaded. Lines 1095 and 1100 verify the code to be edited. If the verification fails, instructions are given in lines 1105-1120. If the verification is successful, the edits are made in either line 1125 or 1130. The file is then unlocked, the edited file saved, and then locked again in lines 1135-1145. Finally, a closing message is printed on the screen in lines 1220 and 1225. If neither file can be found on the disk, and abort message with bell clanging is given in lines 1230 and 1235. The error handling routine begins at line 1200 with a call to a subroutine that repairs the stack. Then, List Handler is identified by a deliberately caused error. If the error is other than error number 6 (FILE NOT FOUND), the error that has occurred is identified in line 1205 along with the line number involved.

## Discussion

The protection used on the Handlers is typical of add-on schemes. The protection is implemented through some protection file during boot-up. Then, if the protection hurdles are all cleared, the unprotected program code is executed. Usually, the reset pointers have been changed to point to some re-boot or some memory wipe routine. This type of protection

can be defeated by eliminating the protection code and going directly to the program code to begin execution. This is what I have done in this softkey.

On the brighter side, the protectors sometime make things easy. Always examine everything on a disk, including the deleted files. About three years ago, I spent hours on the program SIDEWAYS only to discover a complete description of the protection in a deleted file. List Handler has a deleted file named MAIN2 that has not been overwritten. It loads at \$3800. This is the address that MAIN3 jumps to after all the protection checks have been satisfied. If MAIN2 is undeleted and then BRUN; presto, the List program comes right up. A file can be undeleted by using a sector editor or by using a utility with an undelete option. Both Locksmith 6.0 and Copy II Plus have undelete options. Undelete MAIN2 and change the HELLO program to BRUN MAIN2, and you will have a deprotected List Handler the easy way. The other procedure is recommended because this one may not last long.

## Controller

```

1000 REM HANDLERS 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 : D$ = CHR$ (4) : Q$ = CHR$ (34)
1055 ONERR GOTO 1200
1060 Z = 0 : F$ = "WB"
1065 Z = Z + 1 : PRINT : PRINT D$ "VERIFY" : F$ ;
      " ,D" : D2
1070 IF Z = 1 THEN NS$ = "WORD"HANDLER"
1075 IF Z = 2 THEN NS$ = "LIST"HANDLER"
1080 PRINT D$ "BLOAD" : F$
1085 BA = PEEK (43634) + PEEK (43635) * 256
1090 LA = PEEK (43616) + PEEK (43617) * 256
1095 IF PEEK (BA) = 8 AND PEEK (BA + 1) = 104
      AND PEEK (BA + 2) = 141 AND PEEK (BA + 3)
      = 97 THEN 1125
1100 IF PEEK (BA + 256) = 8 AND PEEK (BA + 257
      ) = 104 AND PEEK (BA + 258) = 141 AND
      PEEK (BA + 259) = 97 THEN 1130
1105 PRINT "EDIT*FAILED.*SEARCH*" : NS$ ; ""FOR"
1110 PRINT "THE*HEX*STRING*" : Q$ ;
      "08*68*8D*61" : Q$ ; ""AND*CHANGE"
1115 PRINT "TO*" : Q$ ; "B8*60" : Q$ ;
      ""BEGINNING *AT*" : Q$ ; "08" : Q$ ;
      " ,WRITE"
1120 PRINT "SECTOR* BACK* TO* THE* DUPLICATE*
      DISK ." : END
1125 POKE BA ,184 : POKE BA + 1 ,96 : GOTO 1135
1130 POKE BA + 256 ,184 : POKE BA + 257 ,96
1135 PRINT D$ "UNLOCK" : F$
1140 PRINT D$ "BSAVE" : F$ ; " ,A" : BA ; " ,L" LA

```

```

1145 PRINT D$ "LOCK" : F$ : GOTO 1220
1200 CALL 822 : IF PEEK (222) = 6 THEN 1210
1205 PRINT "ERROR* NO.*" : PEEK (222) : "" IN*
      LINE*NO.*" : PEEK (218) + PEEK (219) * 256
      : END
1210 IF Z > 1 THEN 1230
1215 F$ = "MAIN3" : GOTO 1065
1220 HOME : VTAB 4 : HTAB 6 : PRINT NS$ ; "" NOW*
      DEPROTECTED"
1225 VTAB 6 : HTAB 12 : INVERSE : PRINT "THAT*S*
      ALL* FOLKS" : NORMAL : VTAB 24 : END
1230 FOR I = 1 TO 3 : CALL - 1052 : PRINT SPC (6) :
      "ABORT" : : NEXT
1235 PRINT : PRINT : HTAB 5 : PRINT "SOMETHING*
      IS* WRONG.* CHECK* DISKS!" : END

```

## Controller Checksums

1000	- \$356B	1110	- \$7E4B
1010	- \$2544	1115	- SCF49
1020	- \$08CD	1120	- SE43B
1030	- \$0B00	1125	- \$399A
1040	- \$A1C7	1130	- SF06B
1050	- \$EB03	1135	- \$BAE3
1055	- \$1B38	1140	- \$B201
1060	- \$A05D	1145	- \$FEFE
1065	- \$4EEB	1200	- \$FB62
1070	- \$8DCC	1205	- \$749F
1075	- \$7096	1210	- \$69B8
1080	- \$0129	1215	- \$832B
1085	- \$E307	1220	- \$77D0
1090	- \$E3F0	1225	- \$AC03
1095	- \$E05F	1230	- \$50DC
1100	- \$031A	1235	- \$D6CA
1105	- \$F4FE		

Softkey for...

**PFS software - ProDOS**  
Software Publishing Corporation

*pfs: Write*

*pfs: File*

*pfs: Graph*

*pfs: Report*

*pfs: Plan*

### Requirements

- Apple IIc or IIe, 128K
- One disk drive
- COPYA or Locksmith Fast Disk Backup
- Sector Editor with search capability



The ProDOS versions of the PFS Software shown in the title are much improved over the older DOS 3.3 versions. The individual packages can function independently of each other or as an integrated system.

The *PFS:Dictionary* is a 75,000-word dictionary that is used to check for any misspelled words in a document. It is on an unprotected disk.

Also, supplied is a *PFS:Convert* program that is used to convert previous non-ProDOS PFS data files to ProDOS PFS data files. This program is also on an unprotected disk.

## The Protection

I usually start the protection analysis by attempting to copy the software with *Locksmith 6.0 Fast Disk Backup*. I use this program because it is a good diagnostic tool, and it takes only 25 seconds to copy and verify a disk. It will continue to copy if it encounters non-standard formatting and will indicate the nature of the discrepancy, such as **A** for address field error or **D** for data field error. This is shown by the track and sector so that one knows immediately where to start looking for the modifications. A hard copy of this display can be obtained if desired. In addition, any sector that has an error will be left in a standard formatted condition on the copy.

All of the PFS software will copy with no indication of error. This shows immediately that no formatting modifications have been used. However, the copies thus produced will not complete the boot. Each hangs just after ProDOS is loaded and the display switches to 80-columns.

A word about how ProDOS boots is in order. ProDOS is loaded first. It then loads the first system file it can find into memory at \$2000 and passes control to this file. If this is *BASIC.SYSTEM*, this system file then runs a BASIC file called *STARTUP* to load the balance of the program. In our case the first system file is something like *WRITE.SYSTEM*, for instance, for *PFS:Write*. In the ProDOS environment this first system file nearly always contains the protection code. This is indicated here, because the hang occurs almost immediately after ProDOS is loaded.

If there are no format modifications, the protection code must be looking for a signature that is difficult to reproduce or it must be doing a nibble count somewhere. I have my machine equipped with the *Senior PROM* and with *Trak Star*. *Senior PROM* provides an NMI (nonmaskable interrupt) capability with all the needed subroutines in the PROM. The *Trak Star* shows what track drive 1 is accessing at any given instant. These tools make the task of determining the protection much easier. If a copy of any of the PFS software made with *COPYA*, *Fast Disk Backup*, or any other

standard whole disk copier is booted and NMI'ed just before it hangs; the *Trak Star* indicates track \$00 and the Program Counter from *Senior PROM* shows an address of \$F91 to \$F94 depending on the time of the NMI. The stack shows the first return to be to \$E45. Entering the monitor from *Senior PROM* and disassembling the code around these locations shows the following:

- **1** At \$E42 there is a jump to the subroutine at \$F63. At \$E45 the content of the accumulator is compared with \$00. If the accumulator is not equal to \$00, a failure path is taken. If it equals \$00, program execution jumps to \$107D, the entry point to the program.

- **2** The subroutine at \$F63 is the protection code. It sets up the read block routine and then looks for a signature. It plays some bit-games with what it finds and then returns to the caller with either \$00 or \$FF in the accumulator. The return is at \$FD0. From what was revealed at \$E45, the value of \$00 is the good number to have in the accumulator. This entire protection code can be avoided by going to \$F63 and entering the code to load the accumulator with \$00 and then return to the caller; namely, A9 00 60.

As a matter of interest the signature that is sought by the protection code is contained in the sync field that precedes physical sector \$01 on track \$00. This corresponds to DOS 3.3 sector \$07 or ProDOS block \$04. Use a nibble editor to see this signature.

The sector interleaving is defined in the books *Beneath Apple DOS* and *Beneath Apple ProDOS* published by Quality Software. The nibbles are:

```
Disk Nibbles: FB FF FF FF
Framing Bits: 2 0 2 0
```

The number of framing bits in each byte was determined using *Locksmith 6.0 Framing Bit Analyzer*. The framing bits are used by the deprotection code.

For those that are interested in examining the code discussed above, boot the copy and let it hang. Then, press **[RESET]** at least twice in rapid succession until the BASIC prompt appears. Enter the monitor (**CALL -151**) and disassemble the protection code (**F63LLL**). Then disassemble the code that calls the protection (**E42LL**).

## The Deprotection

- 1** Copy the PFS software using any standard whole disk copier such as *COPYA* or *Fast Disk Backup*.

- 2** Make the following edit (DOS 3.3 interleaving) using your favorite sector editor:

Trk	Sct	Bytes	From	To
\$04	\$0F	\$F5	A0 01 A2	A9 00 60

- 3** Write the change back to disk.

- \*4** If the above fails, search the entire disk for the string: A0 01 A2 06 BD.

- \*5** At each occurrence, change A0 01 A2 to A9 00 60 and write the sector back to disk:

You now have a *COPYA*able backup of the PFS software.

## Remarks

The type of protection employed on PFS software is referred to as the bit insertion method. It is a mild form of protection that defeats standard copiers; however, most bit-copiers can produce a protected backup.

This is the same type of protection used on the earlier DOS 3.3 versions. See *Deprotecting PFS Software* by Gary Wolfe in *COMPUTIST #14*.

*Locksmith 6.0* can be used to automate the softkey provided herein using the Locksmith Programming Language. Last but not least, a Super IOB Controller can be written to copy the software and make the edits. Such a controller is provided at the end of this article for those that have Super IOB 1.5.

As mentioned earlier *PFS:Dictionary* and *PFS:Convert* are unprotected programs.

## Controller

```
380 AS = "VOLUME*NUMBER*FOR*COPY*=>1" : HOME
      : GOSUB 450 : HTAB 33 : INPUT "" : VLS : VL
      = VAL (VLS) : IF VLS = "" THEN VL = 1
1000 REM PFS PRODOS SOFTWARE
1005 REM CONTROLLER
1010 TK = 0 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1020 GOSUB 490 : RESTORE : GOSUB 610 : T1 = TK
      : TK = T1 + 6 : GOSUB 310 : TK = T1
1030 GOSUB 490 : GOSUB 610 : IF PEEK (TRK) =
      LT THEN 1050
1040 TK = PEEK (TRK) : ST = PEEK (SCT) : GOTO
      1020
1050 HOME : AS = "THAT'S*ALL*FOLKS" : GOSUB
      450 : END
2000 DATA 3*CHANGES
2010 DATA 4 ,15 ,245 ,169 ,4 ,15 ,246 ,0 ,4
      ,15 ,247 ,96
```

## Controller Checksums

380	- \$4C64	1030	- \$65C8
1000	- \$81CA	1040	- \$B961
1005	- \$EA67	1050	- \$9F46
1010	- \$8777	2000	- \$FFEB
1020	- \$7CBB	2010	- \$6A2B

† Softkey Addendum for...

## Zardax v5.2.1

Computer Solutions

### ■ Requirements:

- Zardax Program Disk
- Zardax Utilities Disk
- COPYA, Advanced COPYA, or Locksmith 6.0
- COMPUTIST #31
- 2 Blank Disks

Zardax is a word processor based on a modified DOS 3.3 operating system. It is designed to operate on machines with as little as 48K of memory. It is a good word processor for its time (early 1980's). However, it does not measure up to the latest crop of word processors that take advantage of added memory and ProDOS. It needs to be updated to compete.

a note on W. Hinger's...

### Deprotecting Zardax COMPUTIST #31, Page 24

In COMPUTIST #31 William Hinger published a masterful, scholarly softkey for Zardax. The only problem is that the application of this softkey to version 5.2.1 does not produce a backup that can be re-configured with the Utilities disk.

The protection used is implemented through a combination of BASIC and machine language programs, and the disk formatting is modified. The heavy protection is included in the file HELLO1. I never did liberate this Applesoft BASIC program. However, I was able to trace the file from its track/sector list, read the consecutive sectors with INSPECTOR/WATSON, use the ASCII display, and view what was happening.

Fortunately, Mr. Hinger wrote a machine language program called BOOTCODE to function as the second boot stage that loads ZARDAX, obviating the need for HELLO1.

On the original, modified data marks are used on track \$03 sectors \$0F, \$0E, \$0D, \$0C and \$0B. Tracks \$05 — \$10 are written in a non-standard format. An address header of D5 AA AB is used. Most of this is explained in the cited article. In fact, the protection used on version 5.2.1 is identical to that described in the previous article.

Except for a minor change to step 4, Mr. Hinger's softkey can be followed down to step 17 under the "Here's How" section.

The modification to COPYA given in step 4 needs to be changed to prevent the string variable, I\$, from getting scrambled and to allow track \$23 to be formatted.

Add the following BASIC lines to COPYA instead of those given in Mr Hinger's step 4.

```
171 INPUT "STARTING TRACK:" ; ST : INPUT
    "ENDING TRACK:" ; LT : IF (ST > 35) OR
    (LT > 35) OR (ST > LT) THEN 170
172 POKE 770 , LT + 1 : POKE 863 , LT + 1 : IF ST
    > 0 THEN POKE 721 , ST - 1 : POKE 722 , ST - 1
173 PRINT : PRINT : INPUT "FORMAT THIS
    DISK?" ; AS : IF LEN (AS) > 0 AND LEFT$
    (AS , 1) = "Y" THEN 175
174 FT = 1
177 POKE - 16642 , 36 : REM FORMAT TRACK
    35 ($23)
```

I suggest that FASTCOPY from Locksmith 6.0 be used to copy the disk instead of COPYA. The tracks that are not of standard format will not be copied, but they will be left in a standard formatted condition.

In step 17 of the "Here's How" (step 18 of the step-by-step) Mr. Hinger modified the binary file LOWER.WRITER written on the program disk. This file is still on the program disk, but it is no longer used by the SETUP program on the Utilities disk. Examination of the SETUP file on the Utilities disk shows that it loads NLWRITER from the Utilities disk. It is a file very similar to LOWER.WRITER, and it is used for the same purpose. This can be seen in lines 100 and 942 of the SETUP program.

Also, in NLWRITER the data header marks are first modified to read/write protected sectors and are then restored to their normal values. This file will have to be changed to permit the SETUP program to work with the unprotected backup of Zardax.

**17** Insert a backup of the Utilities disk and modify it as follows: (This disk is unprotected.)

```
BLOAD NLWRITER, A$4700
CALL -151
4703:02 02 07 5F 48
4771:38 E9
4780:38 E9
4788:BO
```

To prevent data headers from being changed.

```
47BA:AA
3D0G
UNLOCK NLWRITER
BSAVE NLWRITER, A$4700, L$F6
LOCK NLWRITER
```

The backup will now boot in about half the time of the original. As noted in the previous softkey, there are numerous files on the program disk that are not needed for the unprotected backup. Both backups of the program and Utilities disks can be copied with any standard whole disk copier such as COPYA or Locksmith Fast Disk Backup.

To re-configure the backup program disk, boot the backup Utilities Disk. At the menu press **RESET** to enter BASIC. Insert the

backup program disk, put the backup Utilities disk in drive 2, and RUN SETUP from drive 1. Follow the prompts to complete the setup. There is also a SETUP file on the program disk that simply executes SETUP on the Utilities Disk. The file SETUP on the Utilities Disk could be RUN directly, but much disk swapping would ensue. It is better to do as noted above.

This completes my addendum to Mr. Hinger's excellent softkey.

Softkey for...

## Silent Service

MicroProse

### ■ Requirements:

- 64K Apple II with joystick
- NMI capability
- Sector-editor
- Silent Service

Silent Service is an excellent submarine warfare simulation program. The setting is World War II (WWII). All of the scenarios are built around actual WWII submarine missions. A person must learn rudimentary submarine tactics and must make decisions quickly. As skills improve a higher difficulty level can be selected and more difficult missions attempted. One's performance can be measured against the performance achieved on the actual mission. This is a very interesting program.

The disk has a very sophisticated protection scheme that defeats most attempts to make a backup. A registered owner can obtain a backup from the manufacturer for a fee.

### The Known Protection

An examination of the program disk shows it to be of standard format for tracks \$00 through \$21. This can be determined easily by attempting to copy it with Locksmith 6.0 FAST BACKUP. Track \$22 produces address field errors.

A more detailed study of tracks \$21.25, \$21.5, \$21.75, \$22, \$22.25, and \$22.5 reveals that every track from \$21 to \$22.5 is identical to track \$21. In other words, all of these tracks were written simultaneously with a wide head. It is obvious that the protection is associated with these identical tracks.

Using the Nibble-editor in Locksmith 6.0 shows that the tracks under discussion are empty of data for the most part. Sector \$00 has one data byte near the end of the sector. The next step is to determine how the program is accessing these tracks.

Silent Service can be cataloged from a normal DOS 3.3 environment. The bootup program is a binary file named \ that loads at \$0200 with a length of \$EE bytes.



At this point I decided to study the booting process. I have a *Trak Star* that displays the tracks that drive 1 seeks. In the early boot stage the read head seeks tracks \$21.5 and \$22. If it does not find what it wants the program hangs with a "HARDWARE FAILURE" message.

## Code in volatile memory

The next step is to study the bootup program, \\. Since this file loads at \$200, a slight problem is presented. The first few bytes beginning at \$200 are volatile memory. This area is used as the keyboard buffer. This can be remedied by loading \. at \$2200.

**1** Boot a normal DOS 3.3 disk such as *DOS 3.3 System Master*

**2** Enter the Monitor with CALL -151.

**3** Insert *Silent Service* and type:

**BLOAD \., A\$2200**

**4** Turn on the printer, make a disassembly listing and turn off the printer:

```
1 [P]
2200LLLLL
0 [P]
```

A close look at the disassembly listing shows that this program has two possible outcomes. The check is OK and file *G* is BLOADED then file *A* is BRUN.

The check fails, prints the "HARDWARE FAILURE" message and the program enters an endless loop.

Beginning at offset byte \$37 (\$237) is where the checking code resides. The monitor SEEKABS (\$B9A0) and RDADR (\$B944) routines are set up and used to read track \$21.5 and \$22.

The addresses called in \. are \$F9A0 and \$F944, respectively, indicating that DOS has been moved into upper memory. No data is read, just an address mark.

The track number found in each read is compared to \$21. If they match then everything is OK. This check can be defeated by placing a jump at \$237 to jump over all of the checks.

Search the disk for the hexadecimal string 4C 3D 02 4C C2 02. I found the bytes on track \$03, sector \$00, byte \$3B. Note that the offset byte is \$3B which is \$04 more than \$37 when the file is BLOADED. The first four bytes of any binary file give the load address and length in low/high form. This accounts for the offset on the disk being four bytes greater than that in memory.

This bootup check can be defeated by making the following sector-edit:

Trk	Sct	Bytes	From	To
\$03	\$00	\$3B-	4C 3D 02	4C 9D 02

The above was easy, but it is not all there

is. A continuation of the boot after the above edit brings the program to the main menus where the mission, difficulty level, et cetera are selected.

The disk is then accessed again to load the particular mission selected. Just before the beginning of the play display, the protection tracks are sought again. This time the checks are more extensive with at least every half-track between \$21 and \$22.5 being accessed in a rapid, staccato fashion.

A nonmaskable interrupt (NMI) capability is needed here to break into the program. I use the *Senior PROM*. When NMI-ed while accessing the protection tracks, the Program Counter is generally in page \$97 or \$98 of memory.

A review of the files in *Silent Service* reveals a binary file *P* that loads at \$9500 with a length of \$43A bytes; that is, it occupies memory from \$9500 through \$9939. It should be the file containing the latter checks. I presume the name *P* means "protection".

This file can be examined by using a normal DOS 3.3 environment and BLOADing it at \$8500 to prevent overwriting parts of DOS. The only problem is that garbage is displayed when it is disassembled indicating that the file is encrypted on the disk.

Executable code has to be present when the file is being executed. Consequently, I NMI-ed the program while the protection tracks were being accessed and disassembled the code beginning at \$9500 in memory. I printed out the entire file to study it. This is easy with *Senior PROM*.

If I NMI the program at some other time and disassemble it, the same garbage is displayed as obtained when the file is BLOADED at \$8500.

After hours of study I finally determined by brute force how the file is encrypted on the disk. I haven't found the code that does the decryption while the file is being executed.

The following example will define the method of encryption when BLOADED into memory and the corresponding decrypted bytes when the file is being executed:

Bytes BLOADED:  
01 00 03 02 05 04 07 06 09 08 0B 0A 0D 0C 0F 0E . . . .

Decrypted bytes:  
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F . . . .

It can be seen that the bytes are reordered when decrypted. The decrypted bytes are the executable code. The BLOADED (encrypted) bytes show the order of the corresponding bytes on the disk. Everything is executed in reverse order in much the same way a memory address turns out in reverse order when executed by the 6502 microprocessor.

## Enemy Mine

If the checks in *P* fail, the program continues

but a player will soon "ENCOUNTER AN ENEMY MINE".

Many attempts to edit the code on pages \$97 and \$98 resulted in failure. This is the code where the disk checks are being made. I finally determined that a RETURN near the beginning of *P* will do about as good as any.

Whether or not the carry flag is set or cleared does not matter. Perhaps the condition of the Status Register when the program exits the checks does. Most addressing is done in an indirect manner through page \$00, making it difficult to trace what is happening.

In my efforts I used the feature of *Senior PROM* that moves pages \$00 — \$08 to pages \$20 — \$28 to preserve the volatile memory. It would be instructive for the interested person to study the *P* file and its use of page \$00. The discussion that results is beyond the scope of this softkey.

Moving on with the approach used here, locate the beginning of *P* on the disk. Search the disk for the following hexadecimal string: 08 20 08 95 85 68 60 FB. This is the beginning of *P* which is loaded at \$9500. On my program disk it is located at track \$0E, sector \$0A, byte \$04.

Make the following sector-edit:

Trk	Sct	Bytes	From	To
\$0E	\$0A	\$06-	08 95 85 68 60	A9 95 85 32 60
			?	F7 A9 18 85 02
				60 F9

## Incomplete Crack

The above edit sets up memory location \$F7 and \$F9 to match the values found there after NMI-ing the original when it exits the checking program. It then returns to the caller.

This edit does not produce a complete 'crack' of *Silent Service*, but it does yield a significant 'bend.' The only problem left during the play is that **occasionally after a convoy is attacked all the ships in the convoy will undergo a metamorphosis and change into destroyers.**

**1** Copy *Silent Service* using any whole disk copier that will ignore errors. *Locksmith 6.0 FAST BACKUP* will do this and leave the track/sectors with errors formatted.

**2** Using your favorite sector-editor, I use *Locksmith 6.0 INSPECTOR/WATSON*, make the following edits:

Trk	Sct	Bytes	From	To
\$03	\$00	\$3B-	4C 3D 02	4C 9D 02
\$0E	\$0A	\$06-	08 95 85 68 60	A9 95 85 32 60
			?	F7 A9 18 85 02
				60 F9

The above edits terminate access to tracks \$21 — \$22.5. If the original is copied with *Locksmith 6.0 FAST BACKUP* as indicated



above, track \$22 will be left in a standard format. Both tracks \$21 and \$22 could be used for some other purpose if they were marked 'free' in the VTOC. That is an exercise for someone who needs extra storage space.

## Reflections

As mentioned above the procedure given herein is not a complete 'crack' of *Silent Service*. It does produce a playable backup with an added level of difficulty --- the convoy ships changing into destroyers at certain times. This is exciting to the player. I have discovered that if I keep the submarine surfaced and fight it out with the destroyers using the deck gun, torpedoes, and brilliant tactics... about half the time I can sink all of them.

② I called the previous section "The Known Protection" because I have always found another level of protection when I got one defeated. Presently, I have been unable to locate the code that turns convoy ships into destroyers. I think that this aspect of the protection is unique and my hat is off to the protectors of *Silent Service*. Maybe with what is given here a more perceptive deprotector can find the protection that is still present on the disk. Until then, this softkey produces a backup that one can enjoy.

Softkey for...

## Rocky's Boots 4.0

The Learning Company

### ■ Requirements:

- 48K Apple II
- COPYA
- A Sector-editor
- A blank disk
- Rocky's Boots v4.0

*Rocky's Boots* is an educational program for persons 9 years old and up. It helps to develop skills in logical problem solving, abstract reasoning, and creative thinking. The games that can be played or invented use concepts that are analogous to those used in computer design and machine language programming. These games can be saved to a storage disk for subsequent play or modification. The storage disk can be initialized from within the program. This is an excellent program. Unfortunately, it is placed on a thin disk, and it is copy-protected. Registered owners can obtain a backup disk from the manufacturer for a fee (\$12 plus \$3 for shipping-handling). Since young (and old) users can destroy original programs, I decided to remove the protection so that legal backups can be easily made.

## The Protection

The *Rocky's Boots* disk has a slightly modified format. The address fields and data

fields have epilogues of \$FF \$FF instead of the standard \$DE \$AA. This can be determined by using a nibble-editor. I use the *Locksmith 6.0 DISK EDITOR*. These epilogue changes will prevent any standard copier from copying the disk unless patches are made in memory to accommodate the modifications.

Much can be learned sometime from watching and listening to the boot process. I have a *Trak Star* that shows the tracks that drive 1 seeks. During the initial boot stage, the read/write head swings out to track \$22 and then returns to track \$01. This is a sure indication of some kind of protection being on track \$22.

At this point I decided to normalize the program disk to facilitate working with it. This is easy to do using *COPYA* from the *DOS 3.3 System Master* disk. Do the following:

- 1** Boot the *DOS 3.3 System Master* or any other standard DOS 3.3 disk
- 2** Tell DOS to ignore epilogues and checksum in READ mode:  
**POKE 47426, 24**
- 3** Run *COPYA* from the *System Master* disk.  
**RUN COPYA**
- 4** Follow the prompts and copy *Rocky's Boots* to a blank disk

The backup just made will have normal marks. It will be able to be read by any sector editor without making patches. In general a copy made this way will not boot until the epilogue marks in its RWTS (Read and Write Tracks and Sectors) are changed back to normal. In the case of *Rocky's Boots* this is not true because its RWTS will read either altered or normal epilogue marks, as we shall see.

At this point it is necessary to find and examine the code that controls the checks being performed when the read/write head swings out to track \$22. The easiest way to do this is to interrupt the program while track \$22 is being accessed. This program can be interrupted with a **ⓄRESET** keypress but it is not easy to determine where the program was executing in memory this way. I use the *Senior PROM* to interrupt programs because it can preserve volatile memory, and it can display the Program Counter and the contents of the accumulator and all registers. Also, the stack can be shown easily.

Booting the copy just made shows that it hangs on track \$22 and then re-calibrates and re-boots. An interrupt during track \$22 access shows the Program Counter to be around \$5652 in memory with the first return on the stack at \$4F33. This means that one must be added to this value to get the return address giving \$4F34.

Examination of the code on page \$56 shows

that it is an image of DOS 3.3 page \$B9. Further examination shows that the designers of *Rocky's Boots* have used pages \$B8, \$B9, and \$BA of the DOS 3.3 RWTS and have located them at pages \$55, \$56, and \$57, respectively. Of particular interest are the RDADR routine located at \$5644 through \$569F and the SEEKABS routine located at \$56A0 through \$56FF.

A disassembly of the code will not be given here in the interest of brevity. Anyone can disassemble these codes from normal DOS 3.3, or a **ⓄRESET** keypress during track \$22 access will interrupt the program permitting the above memory addresses to be disassembled.

Of greatest interest now is the routine that calls the above DOS 3.3 routines. This can be seen on page \$4F of memory.

The seek of track \$22 is set up at \$4F10 then the SEEKABS routine is called at \$4F12. More set up work is done in page \$00, the drive is turned on, and the RDADR (read address field) routine is called at \$4F31. Note that this standard routine reads standard address field marks. On return from this routine sector \$0F is sought, then particular bytes are looked for: \$D5, \$E7, \$E7, and finally \$EE. Now, the important part. If the checks fail, the Carry Flag is cleared at \$4F8D and the execution branches to a failure path, re-booting at \$4FD2.

The only thing necessary to deprotect this program is to edit \$4F8D to set the Carry Flag. This will circumvent all of the disk checks. In effect this tells the program that all is well no matter what is found on the disk.

- 1** Use *COPYA*, or a *Super IOB* Controller if preferred, to normalize and copy *Rocky's Boots* as shown above.
- 2** Search the disk for the hexadecimal string 18 90 3E A9 60 8D. This was found on track \$00, sector \$01, byte \$8D on my program disk.
- 3** Make the following edit:

Trk	Sct	Bytes	From	To
\$00	\$01	\$8D	18	38

This procedure produces a completely deprotected backup of *Rocky's Boots*. As an alternative, the controller at the end of this article can be used with *Super IOB 1.5* to accomplish the same task.

## Additional Discussion

As a matter of future interest, another checking routine is found on page \$5D. It is not used! It is almost identical to the checking routine found on page \$4F as described in this article. I suspect that it has been placed there as an obfuscation factor or for use on some future version.





It is interesting to note the nature of the protection employed on track \$22, sector \$0F. Within the data field a large field of E7's is written. This field starts with a sync 50E7 with one framing bit. The next E7 is also a sync byte with one framing bit. Further on E7's with no framing bits and with two framing bits are found.

The disk controller hardware should not lose synchronization but bytes different from E7 should be read. This is a rather sneaky bit-game. Fortunately, it can all be circumvented. The framing bit analysis was done using the *Locksmith 6.0 FRAMING BIT ANALYZER*, a very useful utility.

Parenthetically, a standard copier will not copy the sync bytes within a data field. All of the framing bits will be stripped off. The popular bit-copiers will attempt to reproduce the sync bytes; therefore, track \$22 should be bit-copied to attempt to preserve this protection. It does not matter on the deprotected backups.

## Controller

```
1000 REM ROCKY'S BOOTS CONTROLLER
1010 TK = 0 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1020 GOSUB 490 : RESTORE : GOSUB 170 : GOSUB 610
      : T1 = TK : TK = T1 + 6 : GOSUB 310 : TK = T1
1030 GOSUB 490 : GOSUB 230 : GOSUB 610 : IF
      PEEK (TRK ) = LT THEN 1050
1040 TK = PEEK (TRK ) : ST = PEEK (SCT ) : GOTO
      1020
1050 HOME : VTAB 11 : AS = "THAT'S ALL FOLKS" :
      GOSUB 450 : END
2000 DATA 255 , 255 , 255 , 255
2010 DATA 1*CHANGES , 0 , 1 , 141 , 56
```

## Controller Checksums

1000 - \$356B	1040 - \$DEE7
1010 - \$2544	1050 - \$F0FA
1020 - \$E458	2000 - \$9CC8
1030 - \$B573	2010 - \$806A

Softkey for...

## Walt Disney Card & Party Shop

Bantam Electronic Publishing

### Requirements

- 64K Apple
- NMI capability
- Disk-editor
- Sector-editor
- 1 blank disk
- Card & Party Shop

The *Walt Disney Card & Party Shop* (C&PS) is yet another of those type of graphics programs that Broderbund started with *Print Shop*. This one uses Walt Disney characters and

allows a user to create invitations, tags, place cards, place mats, wrapping paper, banners, signs, awards, tickets, cards, letterheads, memos, postettes, and envelopes. It is a very user-friendly, menu-driven program with pictures of all the graphics displayed in a window as they are considered.

The program is on side A with graphics being on side B. Undoubtedly, future graphics disks will be released.

## The Protection

As with all unknown programs I tried to copy C&PS first with *Locksmith 6.0 FAST BACKUP*. I use this one because it is quick, and it will continue to copy if an error is encountered while indicating the type of error. This procedure showed that all tracks copied without error except tracks \$03 and \$11. On each of those tracks address field errors were indicated for all sectors.

Next, I used the *DISK EDITOR* within *Locksmith 6.0* to examine tracks \$03 and \$11. Track \$03 is a track full of a pattern that repeats every \$40 (64) nibbles. The sync field at the beginning is very short. This is a track that could be used to check a signature or to nibble count. Track \$11 is in standard 16-sector format with standard marks except for the address field epilogue. The standard DE AA is changed to D5 AA.

From the PARAMETER selection in the *Locksmith* Main Menu I patched DOS to ignore the epilogues and then used *INSPECTOR/WATSON* to further examine track \$11. This is the track that normally contains the VTOC (Volume Table of Contents) and the CATALOG for the DOS 3.3 operating system. I looked at sector \$00 first, the VTOC.

Bytes \$01 and \$02 point to the track and sector of the first catalog sector. These bytes were 00 0F which is not normal.

I then looked at sector \$0F of track \$11 which is normally the first catalog sector. The catalog started there in a standard fashion. On the other hand, track \$00 sector \$0F contained code. At this point I edited the copy of track \$11 sector \$00 byte \$01 to be 11 instead of 00 to point to the real first catalog sector.

Escaping to the Main Menu, I used the DOS 3.3 UTILITIES and cataloged the copy of C&PS. It came right up showing a disk full of files.

With this knowledge it was obvious that the operating system of C&PS was nearly standard.

While in *Locksmith* I wanted to check the bootup program. This is usually shown on track \$01 sector \$09 beginning at byte \$75. Moving back into *INSPECTOR/WATSON* and reading track \$01 sector \$09 and then toggling the ASCII display key [A] showed that the bootup program is *HELLO*.

At this point, I left *Locksmith* and booted a standard DOS 3.3 disk. The *DOS 3.3 System*

*Master* will suffice. The idea is to patch DOS to ignore the first address epilogue byte and then take a better look at the *HELLO* file. This is done as follows:

First, patch DOS to ignore the first epilogue byte when reading. (\$B992:EA EA)

**POKE 47506, 234**

**POKE 47507, 234**

Now, let's examine *HELLO*. Load it, turn on the printer, CATALOG the disk, LIST the program, and turn off the printer.

**LOAD HELLO**

**PR#1**

**CATALOG**

**LIST**

**PR#0**

The CATALOG shows a normal file structure of binary files except for the *HELLO* file. A binary file named *START* is included at the beginning of the CATALOG. The listing of the *HELLO* file shows a strange POKE to SFE on the zero page. The next line BRUNs *SSPROT\$\$1* which is a continuation of the protection.

Examination of *SSPROT\$\$1* shows that it is the file that executes the disk-check of track \$03. It searches for a signature and will fail if it is not found.

It concludes by BLOADing *SSPROT\$\$A*. This file is a short file that is a binary load of a BASIC file. It is encoded so that it must be executed to see the real nature of the file. A nonmaskable interrupt capability is needed to examine this file. When this is done, *SSPROT\$\$A* is a file that does three things. It sets MAXFILES to 1, CLOSES all files, and BRUNs *START*.

All that has been done to this point demonstrates that C&PS uses a slightly modified DOS 3.3 operating system. The modifications can be determined as follows:

Boot C&PS and immediately reset into BASIC by pressing [RESET]. If this is not done immediately before the program is loaded the disk will re-boot. The RUN flag is set somewhere.

**Call -151**

Move the modified RWTS (Read and Write Tracks and Sectors) into lower memory out of the way of a re-boot.

**4800<B800.BFFF**

Boot a standard DOS 3.3 disk, reset into BASIC.

6 [P]

[C] or [RESET]

Enter the monitor and compare the altered RWTS with the standard RWTS.

**CALL -151**

**4800<B800.BFFF**



The comparison shows that \$B992.\$B993 has been changed from D0 AE to EA EA. This confirms what has already been determined; namely, the first byte of the address epilogue is ignored in the read mode. The comparison will also show that pages SBB and SBC are different. These pages are buffer areas used by DOS 3.3 and the differences are unimportant.

To summarize, the protection used on C&PS is modified address epilogue marks on track \$11; a disk check of track \$03 for a signature executed from the file *SSPROT\$S1*; and the binary file *SSPROT\$SA* which is in reality a BASIC file disguised as a binary one. This last file sets up the machine for the execution of *START*, the real startup program.

## The Softkey

From the above description of the protection this program can be normalized by transferring the files to a standard DOS 3.3 disk that has been initialized with a null HELLO bootup program. Do the following:

**1** Boot a DOS 3.3 disk, preferably a fast DOS such as *Diversi-DOS*.

**2** Clear memory.

### NEW

**3** Place a blank disk in the drive and initialize it.

### INIT HELLO

**4** Modify DOS to ignore first epilogue byte when reading.

**POKE 47506, 234**  
**POKE 47507, 234**

**5** Remove the initialized disk, insert the *DOS 3.3 Master* disk and *BRUN FID*.

### BRUN FID

**6** Follow the prompts and use the wildcard = to transfer all of the files from the original C&PS to the initialized disk.

**7** Make the following sector-edits to the copy using your favorite editor:

Trk	Sct	Bytes	From	To
\$11	\$00	\$01	00	11
		\$40-	00 00	FF E0
		\$44-	00 00	FF FF

**8** Enter BASIC, key-in, and SAVE the following *HELLO* program to the copy.

**5 D\$ = CHR\$(4)**  
**10 PRINT D\$;"MAXFILES 1"**  
**20 PRINT D\$;"CLOSE"**

## 30 PRINT D\$;"BRUN START"

The above produces a completely deprotected version of side A to C&PS. A further refinement would be to DELETE the files *SSPROT\$S1* and *SSPROT\$SA* to make more room on the disk. However, this is not necessary.

For those that have *SUPER IOB 1.5*, install the Controller at the end of this article to do all the grunt work described above. For the *Super IOB* method a disk must be initialized first with a null *HELLO* file as described above. Again, it is preferable to use a fast-DOS to speed up the loading of the binary files. This will be the copy disk to use when running *Super IOB*. When the copy is done, key-in and SAVE the *HELLO* file at the end of this article to the copy.

Side B is unprotected. It can be copied with any standard whole disk copier such as *FAST BACKUP* or *COPYA*.

## Discussion

As a matter of interest, a protected backup of C&PS can be produced as follows using *Locksmith 6.0*:

**1** *FAST COPY* tracks \$00 — \$22

**2** *BIT COPY* tracks \$03 and \$11 using default values.

Usually the copy of track \$03 will fail or a short track with a length of about \$1600 nibbles will be written. This will fail to produce a successful backup. The length of the track written to the copy must be in the range of \$18F0—\$1900 nibbles.

Time can be saved by entering the *DISK EDITOR* and reading track \$03 into the buffer from the original.

Use the fast movement of the cursor to find the excuse for a sync field.

Set the track start manually at the beginning of this. Set the verify start at the first nibble following this field.

Move back to the track start and move forward \$1900 nibbles in the buffer. The ragged sync field should be near.

Set the track end at the end of the normal nibbles before the sync nibbles start. Write the track back to the copy. This procedure will produce a successful protected backup.

## Disney's Comic Strip Maker

Bantam Electronics is also distributing a companion program called *Walt Disney Comic Strip Maker*. More than likely it has the same or a similar protection on it that C&PS has. If the program is file-based and the DOS is modified somewhat, the files can be liberated in a manner similar to the method used in this softkey. This is especially true if the major protection is contained in files as it is in C&PS.

## Controller

```
1000 REM CARD & PARTY SHOP
1010 TK = 4 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1015 POKE 47506 , 234 : POKE 47507 , 234
1020 GOSUB 490 : RESTORE : GOSUB 610 : T1 = TK
      : TK = T1 + 6 : GOSUB 310 : TK = T1
1030 GOSUB 490 : GOSUB 610 : IF PEEK (TRK ) =
      : LT THEN 1050
1040 TK = PEEK (TRK ) : ST = PEEK (SCT ) : GOTO
      : 1020
1050 HOME : AS$ = "SAVE^ THE^ NEW^ HELLO^ FILE^
      : TO^ THE^ COPY." : PRINT : GOSUB 450
1055 PRINT : AS$ = "THAT^S^ ALL^ FOLKS" : PRINT
      : GOSUB 450 : END
2000 DATA 5^CHANGES
2010 DATA 17 , 0 , 1 , 17 , 17 , 0 , 64 , 255 , 17 , 0
      , 65 , 224 , 17 , 0 , 68 , 255 , 17 , 0 , 69 , 255
```

## Controller Checksums

1000	- \$356B	1040	- \$53AE
1010	- \$2744	1050	- \$FA5D
1015	- \$1DD4	1055	- \$1D75
1020	- \$E0D0	2000	- \$8BC7
1030	- \$F252	2010	- \$53FA

..... Using .....

## Sider Hard Drives, 3 1/2" 800K Drives, and 5 1/4" Drives

in a DOS 3.3 environment

### ■ Requirements

- Apple IIe enhanced
- Sider 10MB Hard Drive
- 3 1/2" 800K Drive with *Universal Disk Controller* (Central Point Software)
- Two 5 1/4" Apple Disk II Drives with *Disk II Interface Card*
- UniDOS Plus* (MicroSPARC)
- DAVID-DOS II.2* (David Data)
- SiderWare* Support Utilities for DOS 3.3

### The Problem

The DOS 3.3 operating system for the Apple II series computers was the standard for a long time. However, ProDOS has gradually become the preferred operating system. This is due to the fact that ProDOS facilitates file manipulations on hard drives, 3 1/2" drives, 5 1/4" drives and combinations thereof.

Conversely, DOS 3.3 does not recognize 3 1/2" drives and does not employ the hierarchal



file structure inherent in ProDOS. However, those that still extensively use DOS 3.3 have a need for a DOS 3.3 modification that will permit file manipulations among the various type drives mentioned above. This need has been partially addressed by the software defined above.

The objective of this article will be to derive a procedure that does permit DOS 3.3 operations among all the type of drives mentioned above in the hardware requirements.

### MicroSPARC's UniDOS Plus

UniDOS Plus, developed by MicroSPARC, solves some of the problem. It is a very compatible modification of DOS 3.3 that recognizes 3 1/2" drives. Utilities are included on the disk that permit more-or-less normal operation with 3 1/2" and 5 1/4" drives in a DOS 3.3 environment.

A modified FID is included on the disk that permits access to drives 3 and 4 installed on the 3 1/2" disks. Surprisingly, this modification does not provide for the independent specification of the volume number.

When the 3 1/2" disk is formatted with the INIT command one 400K volume is placed on side 1 of the disk. Another 400K volume is placed on side 2. Side 1 is accessed as drive 1; side 2 as drive 3. If a second 3 1/2" drive is daisy-chained off of the first; side 1 of it is accessed as drive 2; side 2 as drive 4.

However, with all of these good features, the DOS 3.3 partition of the hard drive cannot be accessed from the 3 1/2" drive.

### David Data's David-DOS II.2

David-DOS II.2, developed by David Data, is a modification of DOS 3.3 that speeds up the loading and saving of all file types.

Twelve commands have been added that improve basic DOS. One of these is HIDOS that moves DOS to the Language Card and continues operations of programs.

A number of useful utilities are included on the disk. An applesoft program named MAKE BRUN-ABLE DOS is of particular interest. Its purpose is to capture the DOS resident in memory as a binary file that is BRUNable. It is set up to capture DOS 3.3 or David-DOS in low memory and David-DOS in high memory. These binary files can be transferred to a hard drive and executed there to install the particular DOS desired.

### SiderWare's Hard Drive Support

SiderWare Support Utilities for Apple DOS 3.3, provided with the Sider hard drive, provides several very useful utilities.

A modified FID is included that does permit the specification of volume number as well as

slot and drive numbers. A patch to the program DOS FILE UTILITY is given on page 12-8 of the latest manual to defeat slot checking. This utility contains the modified FID program. The Technical Assistance Personnel give an addition to this patch for those using the Universal Disk Controller (UDC). For clarity the patch is repeated here:

**UNLOCK DOS FILE UTILITY,V1  
BLOAD DOS FILE UTILITY  
CALL -151  
4812:EA EA EA N 4816:94  
51A7:07 for UDC  
E003G  
BSAVE DOS FILE UTILITY,AS4800,L\$15A6  
LOCK DOS FILE UTILITY**

The Sider supports other operating systems in its DOS 3.3 partition such as Pronto-DOS, Diversi-DOS, and David-DOS and provides programs or instructions for the installation of these DOS modifications.

Of course, DOS 3.3 is the default operating system. If other modifications of DOS 3.3 can be installed on the Sider, why not UniDOS Plus? If possible, this might solve the access problem previously described.

### The approach

The plan is to get UniDOS Plus into memory by booting the UniDOS Plus disk from the 3 1/2" drive.

Insert David-DOS II.2 in the 5 1/4" drive and RUN MAKE BRUN-ABLE DOS.

Answer the prompts and capture UniDOS Plus as a binary file. The program will think that UniDOS Plus is DOS 3.3. The file that is captured is called LO DOS 3.3. It should be renamed LO UNIDOS+ on the disk and copied to the Sider drive and volume desired.

Also, transfer the HELLO file from the UniDOS Plus disk to the 5 1/4" disk, rename it HELLO.UNIDOS+, and copy it to the same drive and volume on the hard drive.

At this point these files cannot be copied with UniDOS Plus resident in memory. It is best to reboot into the Sider and use its support utilities to copy the files from the 5 1/4" drive to the desired Sider drive and volume.

Theoretically, the binary file, LO UNIDOS+, can be BRUN to install UniDOS Plus on the hard drive. When this is done the hard drive does have UniDOS Plus resident, and it works fine except for accessing the 3 1/2" drive.

The plan seems to have failed. After much frustration and experimentation, UniDOS Plus was installed on the hard drive and persuaded to access all drives. This was done as follows:

After the above steps were completed, the 3 1/2" drive was accessed by issuing the PR#x command with the UniDOS Plus disk in the drive. (x is the slot of the 3 1/2" drive.) This warm boots the 3 1/2" drive from the hard drive.

As soon as the booting process is complete and the BASIC prompt | appears, issue another PR#y command. (y is the slot number of the hard drive.) This warm boots the hard drive into the DOS 3.3 partition. BRUN LO UNIDOS+ again.

When this is done all drives can be accessed and files manipulated among them. I do not know why the round trip to the 3 1/2" drive is required, but it works.

I have discovered that the reverse process will work also. In other words, start by booting UniDOS Plus on the 3 1/2" drive. Make the round trip to the hard drive in a manner similar to that previously described and everything will work fine. Apparently, this process initializes the interface cards some way.

### Step-by-step

The following is a step-by-step procedure for successfully installing UniDOS Plus in the DOS 3.3 partition of the Sider hard drive:

**1** Boot UniDOS Plus from the 3 1/2" drive. This gets UniDOS Plus resident in memory.

**2** Insert David-DOS II.2 in the 5 1/4" drive (S6, D1) and:

**RUN MAKE BRUN-ABLE DOS,S6,D1,V254**

Remove the David-DOS disk and insert another DOS 3.3 disk to save the binary file on.

Follow the prompts. Answer Y to the question about creating LO DOS 3.3.

Enter the slot number of your hard drive at the proper time.

Enter 0 for the volume number.

Enter HELLO.UNIDOS+ as the Hello file name.

Enter the type of Hello program: BASIC.

Choose the hard drive.

Then, press the final **(RETURN)**.

When the process is complete, LO DOS 3.3 will have been saved to the DOS 3.3 disk and the BASIC cursor | will appear.

**3** Now:

**RENAME LO DOS 3.3,LO UNIDOS+**

**4** Load the Hello program from whatever Drive x your 3 1/4" drive is in. Mine's in slot 4.

**LOAD HELLO,Sx,D1,V254**

**5** and:

**SAVE HELLO.UNIDOS+,S6,D1,V254**

**6** Now the warm-boot the hard drive, with x as its slot.

**PR#5** My hard drive is in slot 5

Follow the menu to get the DOS FILE UTILITY up and running.



**7** Transfer *HELLO.UNIDOS+* and *LO UNIDOS+* from S6,D1,V254 to S5,D1,V1.

**8** Finally:

**BRUN LO UNIDOS+,S5,D1,V1.**

*UniDOS Plus* is now installed from the hard drive. It will work with the hard drive and the 5¼" drives.

It will not yet access the 3½" drive.

**9** Warm-boot the 3½" drive with the *UniDOS Plus* disk in the drive:

**PR#4**

**10** At the cursor:

**PR#5**

...And warm boot the hard drive again and boot into DOS 3.3:

**PR#5**

**PR#6**

**11** And again...

**BRUN LO UNIDOS+,S5,D1,V1**

At this point *UniDOS Plus* will function normally with all drives.

The *Sider Support Utilities* can be used to perform all file manipulations with any combination of disks.

After *UniDOS Plus* has been captured as a binary file and the files have been copied to the hard drive, the other steps in the procedure can be simplified as follows:

A) Modify the *HELLO* program on the *UniDOS Plus* disk to provide the option of returning to BASIC or of warm booting the hard drive.

B) Similarly, modify *HELLO.UNIDOS+* to provide the options of continuing with *Sider* operations, warm booting the 3½" drive, or returning to BASIC.

## Discussion

The objective of defining a procedure for performing normal DOS 3.3 operations using any mix of the drives identified in the Hardware Requirements section has been laboriously achieved.

It works using the off-the-shelf software described.

Apparently, the patch to *DOS FILE UTILITY* provided by First Class Peripherals does not work with the UDC from Central Point Software. The necessity of making the round trip of warm boots between the hard drive and the 3½" drive to initialize the interface cards is puzzling. Certainly the technical knowledge of how to accomplish the disk access is known because it is routinely done in ProDOS.

I would suggest that one of the manufacturers

discussed herein solve the drive access problem defined in this article in a more straightforward manner.

While this procedure works for the specific hardware system setup described, it will most likely work with other setups.



**Willem Moolenaar**

*Softkey for...*

## Commando

Data East

**1** Make DOS ignore end marks and checksums.

**POKE 47426, 24**

**2** Run *COPYA* from your DOS master disk.

**3** Make the following sector edits:

Trk	Sct	Bytes	From	To
\$00	\$06	\$4B	D0 F4	EA EA
		\$54	D0 36	EA EA
		\$5D	D0 2D	EA EA
		\$70	D0 F4	EA EA
		\$7B	D0 0F	EA EA

*A.P.T for...*

## Commando

Data East

## Unlimited Men & Grenades

The first edit is for unlimited "Arnold Schwarzeneggers" (men) and the second is for unlimited grenades.

Trk	Sct	Bytes	From	To
\$1C	\$09	\$52-	E9 01	EA EA
		\$79-	E9 01	EA EA

*Softkey for...*

## Infiltrator II

Mindscape

**1** Make DOS ignore end marks and checksums.

**POKE 47426, 24**

**2** Run *COPYA* from your DOS master disk.

**3** Make the following sector edits:

Trk	Sct	Bytes	From	To
\$00	\$03	\$42	38	18
\$00	\$08	\$AF	A9	60

*Softkey for...*

## Street Sports Baseball

Epyx

**1** Make DOS ignore end marks and checksums.

**POKE 47426, 24**

**2** Run *COPYA* from your DOS master disk.

**3** Make the following sector edits:

Trk	Sct	Bytes	From	To
\$00	\$05	\$43-	D0 F4	EA EA
		\$4C-	D0 4A	EA EA
		\$55-	D0 41	EA EA
		\$68-	?	EA EA EA EA EA EA A9 FC
				85 F0 85 F3 A9 EE 85 F1
				85 F2

*Softkey for...*

## Championship Wrestling

Epyx

**1** Make DOS ignore end marks and checksums.

**POKE 47426, 24**

**2** Run *COPYA* from your DOS master disk.

**3** Make the following sector edits:

Trk	Sct	Bytes	From	To
\$00	\$05	\$43-	D0 F4	EA EA
		\$4C-	D0 3E	EA EA
		\$55-	D0 35	EA EA
		\$68-	D0 F4	EA EA
		\$7C-	A5 F4	A9 E7

*Softkey for...*

## Portal

Activision

Trk	Sct	Bytes	From	To
\$11	\$01	\$B8-	49 AA	A9 55



Softkey for...

## Aliens Activision

Trk	Sct	Bytes	From	To
\$0B	\$0E	\$9D-	49 AA	A9 55

Softkey for...

## Tass Times in Tonetown Activision

Trk	Sct	Bytes	From	To
\$01	\$0A	\$A4-	49 AA	A9 55

Softkey for...

## GFL Championship Football Gamestar

Trk	Sct	Bytes	From	To
\$1B	\$00	\$96-	49 AA	A9 55

Softkey for...

## GBA Championship Basketball Gamestar

Trk	Sct	Bytes	From	To
\$00	\$0A	\$96-	49 AA	A9 55

Softkey for...

## Manic Mansion Lucasfilm

Trk	Sct	Bytes	From	To
\$00	\$0C	\$98-	49 AA	A9 55

The above (scanning for 49 AA) will most probably work for other new Activision and Activision distributed games!

T. F. Blake

Softkey for...

## Bard's Tale II Electronic Arts

1 Copy disk with Copy II Plus.

2 Get out a sector editor and make the following changes.

Trk	Sct	Bytes	From	To
\$01	\$0C	\$00-	4C 69 05	18 60 DD
\$01	\$0F	\$00-	4C 69 A0	18 60 DD
		\$69-	4C 69 A0	18 60 DD

It is now COPYA-able.

William O. Romine

## Capturing Locksmith FastCopy For IIGs

a note on Jerry Lewis'...

Capturing... Locksmith 6.0 Fastcopy  
COMPUTIST #43, Page 12

After reading Mr. Lewis's article I immediately followed his instructions and made a working binary file of *Fastcopy 6.0* but it would not run on my IIGs but ran fine on a //e.

Shortly after this I received a modification patch from Alpha Logic for my *Locksmith* that fixed *Fastcopy* where it would run on the IIGs - well and good. But a binary file made by Mr. Lewis's procedure would not run on the IIGs.

I then made a copy of *Locksmith STARTUP* and *FASTCOPY* by Mr. Lewis's method outlined in his article. I then compared them to the same files from the binary file that would run on the //e.

Comparison was made by entering the monitor, bloading the file *FC* (IIGs) at \$4000, BLOADing the file *FC* (//e) at \$6000 and doing a monitor compare. The following bytes were found to be different.

address:	FC IIGs	FC //e
\$405A:	1D	08
\$4FBC:	18	EE
\$4FBD:	90	D0
\$4FBE:	35	07

After studying these changes, I found that the only change that seemed to matter was the (08) to 1D. This caused a jump to \$1D00. For there to be the proper file at \$1D00 the rest of *FC* had to be included in the binary file and *LS START* would move it to the proper position. So the boot procedure was followed and *FC* was saved.

SAVE FC, A\$2000, L\$18FD

The above memory comparison was made for *LS START*. There were only 8 bytes different. They were all from \$4003 to \$400B. *FC* was longer now and changes had to be made in *LS START*. The memory move was changed to move in the new length of *FC*. A couple of extra

bytes were included in the final *Fastcopy* binary file beyond the necessary file length.

Two bytes were changed in the move routine in *LS START* so that all of the new *FC* file would be moved. Follow Mr. Lewis's instructions but use the following text file with its changes. (Be sure to add the necessary length to the *FC* file when you copy it).

```

MON C,I,O
CALL -151
0800:22 N 0801<0800.9000M
BLOAD LS START,A$4000
BLOAD ONE SIX,A$4484
BLOAD FC, A$5000
2014<400C.4011M
201A<402F.4034M
2025<4051.405DM
2035<4160.4330M
2206<5000.503FM
2246<503D.505AM
2264<4484.44A3M
2284<50FE.68FDM
2000:A0 00 98 99 00 0C C8 D0
2008:FA EA EA EA EA EA EA EA
2010:EA A9 EA 78
201B:35 20
201E:24 21 AD E9 B7 85 6D
2032:4C 06 22
2045:76 20
2086:09 21
20A9:0B 21
20CA:0B 21
20E7:09 21
20FA:4C 20
20FF:76 20
2104:76 20
2107:47 20
2141:CF 21
220D:71 22
2244:83
2246:3E A0 84 84 3C A0 00
224E:42
2254:22
2258:3A
2273:64 22
BSAVE FASTCOPY,A$2000,L$1B00
3D0G
NOMON C,I,O

```

Note: Have the necessary files on the same disk as this text file before you EXEC it. This text file presupposes that the *Fastcopy* mode to a Binary File had already been modified for the //gs. If a //e copy is to be changed one would need to add to the text file:

2263:1D  
3142: 18 90 35

I have not tested this latter, but it should work to convert a //e to a IIGs binary file.

? Finally, I have not yet figured out how to get the file to read the extra memory in the //gs. If someone has the answer I am waiting.



Edward Teach

Softkey for...

## The Eidolon

Epyx

### Requirements:

- COPYA
- Blank disk
- sector-editor

I am sure that anyone who has deprotected a piece of software has heard "How did you do it?". That question is usually asked by the person who equates computing with scoring a zillion points in *Zaxxon*. I was that person just four months ago. After all, I had a sector-editor and had heard of assembly code.

Deprotection had to be easy, right? For the last four months, I have spent my waking hours with several items. These were *Hardcore Computist*, *Beneath Apple DOS*, *Assembly Lines* and my trusty II Plus. In that time I managed to free a few minor items.

Most of them were deprotected simply by using Ray Darrah's *Super I.O.B.* (nice piece of work Ray). Changing an epilogue here and a prologue there was all that was necessary. If you know what an epilogue and a prologue are then give yourself 5 cracking points. The rest of you go back and reread old copies of the *COMPUTIST*.

So after four months I now know the answer to the "How did you do it?" question. The answer is that it is easy to liberate software. The hard part is obtaining the information and the knowledge to make it easy.

Just as with the master crackist who makes his skills look simple, they were developed through hard work and training. So now that I have advanced from the novice to the rookie level, here is what I have learned:

**1** You must make a *COPYA* version of the disk. To do this, **RUN COPYA** from your DOS master disk. When the disk stops, press **⌘C** to stop the program. Delete line 70 and defeat DOS error checking.

**70**  
**POKE 47426,24**  
**RUN**

It is difficult to examine a protected disk. Making the *COPYA* disk usually entails setting the prologues and the epilogues to a normal format. The *Super I.O.B.* is also excellent for this purpose.

**2** Boot your *COPYA* version. If it

runs then congratulations. If it doesn't run go to step three.

**3** What does the disk do during the boot? Does it boot then reboot then reboot...? Does it boot to the title page then hang up? Does it appear to access the disk?

If the disk keeps rebooting then try searching for code that jumps to memory location \$C600 (the reboot code) or, if the disk spins before hanging up try to locate the code C08C or C08E these locations must be accessed before the disk can be read. If you get the title page then the program hangs up, try to locate C057. This location turns on the hi-res screen. If the code runs until a keypress is needed, then search for C000. This is the location a program checks to see if a key was pressed and to get that key's value.

**4** Remember that addresses in the Apple are stored with the Lo-byte first. I.E. \$C600 is stored as 00 C6 in memory. Also, some programs refer to the disk controller I/O locations directly instead of indexing from \$C08X. A direct read access to a disk controller card in slot 6 would be \$C0EC or EC C0 in memory..... RDEXed

**4** Disassemble the code and try to find what needs modification. By noticing what happens when you boot your *COPYA* version gives a clear indication as to what to search for on the disk. From there you can begin your disassembly.

As a practical example of what all of this means... on to the deprotection.

*The Eidolon* is a quality software game from Lucasfilm and Epyx. One of the more interesting features is that as you advance through the tunnels the hi-res screen scrolls smoothly and maintains perspective (an excellent programming technique).

The protection scheme involves a change in both the address and data epilogue marks. Instead of the normal DE AA, Epyx uses FF FF. This is an easy task for the *Super I.O.B.* to reverse.

Make a *COPYA* version with the correct epilogue marks and boot the copy. You will immediately notice that your disk drive has gone to reboot heaven. Press **RESET** to stop the drive.

Evidently the people at Epyx decided that since they went to the trouble of changing the epilogues, then they might as well check them during the boot. If the marks have been changed then the disk reboots, checks again then reboots...

Interestingly, it would appear that by rebooting Epyx expects the marks to magically transform back to the FF FF that they expected. If you have ever been in the monitor and typed C600G, you know that this will boot drive 1.

Knowing this, I decided to scan the disk for a jump to location \$C600. Hopefully, this would be coded as either 4C 00 C6 (Jump \$C600) or as 20 00 C6 (Jump to subroutine \$C600).



Don't forget to check for an indirect jump to \$C600. It looks like this:

*JMP (\$XXXX) where XXXX is a location in memory that contains the actual address you want (\$C600)*

```
300: 6C 04 03   JMP ($0304)
304: 00 C6.....RDEXed
```

Both the *INSPECTOR* and the *Bag of Tricks* will scan the disk for these specific bytes. I assumed that since the reboot was instantaneous, then the code I needed to locate would be in the DOS image on tracks \$00-\$02. A scan of the disk reveals that track \$01, sector \$0F, bytes \$05,\$06 and \$07 contain the code 4C 00 C6.

This sure looked like the reboot code to me. My first thought was to replace the 4C 00 C6 with EA EA EA. This would tell the computer to perform no operation at that location. I changed the three bytes to EAs and wrote them back on the disk.

Then I typed **PR#6**. The drive made several funny noises and I waited for the reboot. Instead, I was given the title page and sent off to explore the caves.

Now I had a *COPYA* version of a really good game. The only difference between the original and my copy was 'protection' and a few extra head clacks. The extra clacks on my copy bothered me. Did I miss something important?

A disassembly of the code revealed:

```
0800: 20 FD 0E   JSR $0EFD
0803: 90 03     BCC $0808
0805: 4C 00 C6   JMP $C600
0808: 20 14 0B   JSR $0B14
```

Evidently line \$0800 jumps to a subroutine that checks the epilogue marks. If they have been altered (which we did) then the carry bit is set and the code returns via a RTS (return to subroutine) to \$0803. This line says that if the carry is clear then jump over the reboot code.

When we changed line \$0805 to EAs we failed to clear the carry bit. This could cause problems if Epyx later checks the carry bit. Instead we can make line \$0805 clear the carry bit, then the code at \$0803 will not matter. To do this we change \$0805 from 4C 00 C6 to 18 EA EA.

Trk	Sct	Bytes	From	To
\$01	\$0F	\$05-07	4C 00 C6	18 EA EA

The 18 clears the carry bit and then falls into the branch to the start of the game at \$0808. The disk will no longer reboot since that code does not exist.



A disassembly would now show:

```

0800: 20 FD 0E JSR $0EFD
0803: 90 03 BCC $0808
0805: 18 CLC
0806: EA NOP
0807: EA NOP
0808: 20 14 0B JSR $0B14

```

Writing this to the disk gives only one head 'clack' (like the original). This softkey works on this disk.

However, I should stress that I did not disassemble any of the code at \$0EFD (the code which verifies the epilogues). Luckily, Epyx decided to allow the code to simply check the epilogues then return to \$0803. They could have easily stored a number in a certain memory location and then checked during the game to see if the number matched what they placed there. If the numbers did not match then it would be obvious that tampering had occurred and the game could be sent off in some endless loop.

This is why we did not simply remove the jump to the offending code. It seems easier to clear the carry bit then to decipher and rewrite the "check the epilogue code".

So when you deprotect your copy of *The Eidolon* trace the code. For the beginner it will be well worth your trouble.

To trace the code, make a *COPYA* version of your original. Using your sector-editor change track \$01, sector \$0F, bytes \$00-\$02 from 20 FD 0E to 00 00 00. The 00 byte is a software BREAK instruction and will force the program to stop. You should end up in the monitor.

Type *EFDL* and press **RETURN**.

You're on your own now. As for me, I'm off to explore the caves.

Softkey for...

## The American Challenge

Mindscape

By setting the epilogues to a normal format, scanning the disk for "C600", then disassembling the sector, it is easy to see that changing a few bytes will deprotect the disk.

Trk	Sct	Bytes	From	To
\$00	\$0B	\$88-8C	D0 F8 4C 00 C6	EA EA 4C DB 43

See, it gets easier all the time.

Softkey for...

## Saracen

Datasoft/ Intellicreations

This is a nice arcade game, but unfortunately protected. The epilogues and prologues are

normal, but the checksum of the data header and the address header were wrong. Also the volume number on each track was 44. The *Super IOB* will easily make a *COPYA* version. (Which is always my first step when deprotecting a disk). Booting the *COPYA* version causes 'eternal spin'. The secret to this crack lies in the IOB. (If you don't know what an IOB is, then you need to buy a copy of *Beneath Apple DOS*).

Anyway, the byte to change in the IOB is located on track \$00, sector \$01, byte \$10. Change this byte from a 44 to a 00. Write this back to the disk and enjoy.

Trk	Sct	Bytes	From	To
\$0	\$1	\$10	44	00

As a side note you may want to boot up a sector editor and read the disk. The author not only put his name in the sectors, but also his home address (in France) and his phone number. If he did not want you to break this disk and let him know then why bury your name on a protected disk?

Softkey for...

## Bop & Wrestle

Mindscape

Usually when I am working on a disk I keep good notes. The first rule of deprotection - know what you did. So I broke the rule. The epilogues are all FF FF. (Header and Address). The *Super IOB* can fix this.

But what the author also did was to change the read table - located at \$BA96-BAFF. So you have to boot the original disk, break into the monitor and copy the read translate table so you can place it into the *Super IOB*.

Like I said, I broke the rule and did not write this down. Either print it out or write it down.

Now boot up *Super IOB* and exec in your controller. Enter the monitor and type in the read translate table.

When you are done typing press **ESC** and type *RUN*. This will deprotect the *Bop & Wrestle* original.

Also, if you decide to abuse the hi-res banner I would appreciate a "Thanks to Edward Teach".

Softkey for...

## Top Fuel Eliminator

Gamestar/Activision

I love disks that are *COPYA*able but the copy will not run. This usually means that the protection is a nibble count of some kind. To locate the nibble routine search the disk for C08C. On *Top Fuel Eliminator* (TFE) these

bytes show up in the DOS image (tracks 0-2) and also in track 16 sector E. The code here shows:

```

08A8- BD 8C C0 LDA C08C,X
08AB- 10 FB BPL $08A8
08AD- 38 SEC
08AE- 2A ROL
08AF- 25 FC AND $FC
08B1- 49 AA EOR #$AA
08B3- 8D D4 30 STA $30D4

```

What this routine does it to locate a specific nibble, manipulate it and store the final value at \$30D4. While I didn't search the disk, there is code which accesses that location.

To find the value stored at \$30D4 I placed a break 00 at location \$08B3, then booted the disk. The 00 will cause the program to stop and also it will display the contents of the registers.

After the beep, I pressed **RESET** to clear the hi-res screen and was shown that A = 55. The patch needed was to replace the code at \$08B1 (49 AA) with the new code (A9 55).

Gentlemen... start your engines.

Trk	Sct	Bytes	From	To
\$16	\$0E	\$B1-B2	49 AA	A9 55

Softkey for...

## Sub-Mission

by Mindscape

It was like a challenge from Mindscape. The message on the package stated:

**"WARNING: Make a wrong move and one of the captives dies and is erased from the disk...forever!"**

But on the inside was a petition to the "Supreme Commander" who could (for \$7.00 and your original disk) bring the erased captives back from the dead.

What a great marketing idea this is, a disk that tells you it destroys itself. Then for only seven bucks a pop you can have all the replacements you need.

## The Crack

**1** Make DOS ignore end marks and checksums.

**POKE 47426, 24**

**2** Run *COPYA* from your DOS master disk.

**3** Make the following sector edits:

Trk	Sct	Bytes	From	To
\$05	\$0C	\$23-27	C6 F4 D0 94 38	A9 A0 85 F4 18

With these changes the program will boot and run.

A.P.T. for...

## Sub-Mission Mindscape

Track \$22 sector \$00 bytes \$00-08 contains your game status.

You have to win at least one game before you can play with either of the 'captives'. (Remember .. the ones erased from the disk forever!).

The byte patterns are:

**Startup (if you've never won a game)**

0101FB2844010100

**Won one game - no one dead**

0101FF2844010100

**Won one game - Sigourny dead**

0001FD2B44010100

**Won one game - Peter dead**

0100FD2847010100

So Mindscape... erased forever...

Softkey for...

## Facemaker Spinnaker

This one was too simple...

**LOAD LOADER  
100 CALL 24610  
SAVE LOADER**

Softkey for...

## Kung-Fu Master Data East

**1** Make DOS ignore end marks and checksums.

**POKE 47426, 24**

**2** RUN COPYA from your DOS master disk.

**3** Make the following sector edits:

Trk	Sct	Bytes	From	To
\$00	\$05	\$24-36	?	C6 50 F0 03 4C 1E BF A0 00 B1 08 99 00 00 88 D0 F8 38 60

Write these changes back to the disk and enjoy.

You might also want to scan the disk for track/sector lists.

The hi-res banner is on the disk 3 or 4 times, with minor modifications. But the best picture is never called, except in my version.

Softkey for...

## Championship Boxing Sierra On-Line

This disk is COPYAable as-is, so all we have to do is take out the nibble/signature check. The check lives on track \$17, sector \$05.

Trk	Sct	Bytes	From	To
\$17	\$05	\$E3-E5	20 00 1D	EA EA EA

Softkey for...

## Street Sports Baseball Epyx

Fred and I cracked this over the phone one evening, and thought that you would like to be able to back up your copy.

First of all, the disk is COPYAable, but a copy will not run. Therefore, the likely protection is a nibble count of some sort.

Start by scanning the disk for the bytes \$C08C. These bytes show up in a number of places.

But the code we want to alter resides on track \$00, sector \$05. How did we know that this was the code we needed to change?

Well, normally in DOS after the \$C08C is coded there is a search for the bytes D5 AA 96 or DE AA are checked. However in this instance the bytes searched for were E7 and EE.

Reading the code reveals that after these odd bytes are found then subsequent bytes are stored at memory locations \$F0-FF. So all we have to do is store the bytes read and then jump over the nibble count routine.

So take out your trusty sector editor and read track \$00, sector \$05. Then starting at byte XX 21. (XX = the buffer that you are working in) enter:

Trk	Sct	Bytes	From	To
\$00	\$05	\$21	?	A9 FC 85 F0 85 F3 85 F4 85 F5 85 F6 A9 EE 85 F1 85 F2 85 F7 A9 F6 85 F8 A9 00 85 FA 85 FE 85 FF A9 08 85 F9 A9 FE 85 FB A9 A0 85 FC A9 79 85 FD 4C 7A BB

That's it.

# MOST WANTED Softkeys

- ABM Muse
- Accolade Comics Accolade
- Agent U.S.A. Scholastic
- Airheart Broderbund
- Algebra Davidon & Associates
- Apple Super Pilot Apple
- Ballblazer Epyx
- Balance of Power Mindscape
- Bandits Sirius Software
- Bank Street Filer Broderbund
- Brain Bank The Observatory
- Captain Goodnight Broderbund
- Car Builder Optium Resource
- Championship Baseball Activision
- Colossus IV Firebird
- Create with Garfield DLM
- Cross Clues Science Research
- Crossword Magic Mindscape
- Crypt of Media Sir Tech
- DB Master V4.0 Stoneware
- Deep Space Sir Tech
- Earth Orbiting Station (EOS) Electronic Art
- Fay: The Masked Woman Digitech Software
- Fay's Word Rally Digitech Software
- Fay: Word Hunter Digitech Software
- Fun Bunch Unicorn
- Gemstone Healer SSI
- GoldFinger Mindscape
- Goonies Datasoft
- GradeBusters I-2-3 Gradebusters
- Gutenberg Jr. & Sr. Micromation LTD
- Handicapping System Sports Judge
- Hardball II (gs) Accolade
- J & S Grade Book J & S Software
- Jet V I.0 Sublogic
- Jigsaw Microfun
- Le Francais Par Ordinateur D.C. Heath & Co.
- Mathblaster Davidson & Associates
- Microzine 22 & 23 Scholastic Software
- Mind Mirror Electronic Arts
- Movie Monster Game Epyx
- Odin Odesta
- Operation Frog Scholastic Software
- Peeping Tom Microlab
- Personal Finance Manager Apple Computer
- Prime Plotter Primesoft Corp.
- Print Master Unision World
- Quiz Castle Digitech Software
- Silent Service II (gs) Microprose
- Snoggle Broderbund
- Stellar 7 Penguin Software
- Success with Typing VI.2 Scholastic Software
- Super Factory Sunburst
- Think Tank Living Video
- Tower of Myraglen II (gs) PBI Software
- Type! Broderbund
- Visiblend Microlab
- Where in USA is C. San Diego Broderbund
- Work Force II Core Concepts
- The Works First Star Software
- Zorro Datasoft



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

# The Book(s) of Softkeys

## Volume I Compiled from issues 1-5



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

## Volume II Compiled from issues 6-10

contains softkeys for: •Apple Cider Spider •Apple Logo •Artist Arcade Machine •Bank Street Writer •Cannonball Blitz •Canyon Climber •Caverns of Freitag •Crush, Crumble & Chomp •Data Factory V •DB Master •The Dic\*tion\*ary •Essential Data Duplicator I & III •Gold Rush •Krell Logo •Legacy of Llylgamyn •Mask Of The Sun •Minit Man •Mouskattack •Music Construction Set •Oil's Well •Pandora's Box •Robotron •Sammy Lightfoot •Screenwriter II v2.2 •Sensible Speller 4,4c,4.1c •Spy Strikes Back •Time Zone v1.1 •Visible Computer: 6502 •Visidex •Visiterm •Zaxxon • software for: •Hayden •Sierra Online  PLUS the ultimate cracking program: Super IOB 1.5 \*\*\*and more!

## Volume III Compiled from issues 11-15

contains softkeys for: •Alien Addition •Alien Munchies •Alligator Mix •Comp. Prep. SAT •Cut & Paste •Demolition Division. •DLM software •EA (Electronic Arts) software •Einstein Compiler 5.3 •Escape From Rungistan •Financial Cookbook •Flip Out •Hi-res Computer Golf II •Knoware •Laf Pak •Last Gladiator •Learning With Leeper •Lion's Share •Master Type 1.7 •MatheMagic •Minus Mission •Millionaire •Music Construction Set •One-on-one •Penguin software •PFS software •The Quest •Rocky's Boots •Sabotage •Seadragon •Sensible Speller 4 •Snooper Troops II •SoftPorn Adventure •Stickybear series •Suicide •TellStar •Tic Tac Show •Time Is Money •Transylvania •Type Attack •Ultima III Exodus •Zoom Graphics •Breaking Locksmith 5.0 Fast Copy  PLUS feature articles on • Csave • The Core Disk Searcher • Modified ROMs.

Name _____ ID# _____	<input type="checkbox"/> Volume I - \$7.95 + \$2 shipping/handling
Address _____	<input type="checkbox"/> Volume II - \$12.95 + \$2 shipping/handling
City _____ State _____ Zip _____	<input type="checkbox"/> Volume III - \$17.95 + \$2 shipping/handling
Country _____ Phone _____	<input type="checkbox"/> All 3 volumes! - \$30.00 + \$2 shipping/handling
  _____ Exp. _____	
Signature _____ CP50 _____	

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

Send to: Book Of Softkeys PO Box 110846-T Tacoma, WA 98411 (206) 474-5

# COMPUTIST

Note: Softkeys with (†) refer to directions in another COMPUTIST issue or is incomplete as published in this issue.

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

**48** *Features:* Dungeon Editor & Encounter Editor for Ultima III • APT for Shadowkeep • Softkey for Shadowkeep • Softkey for Apple Business Graphics • *RDEX Softkeys:* • \$16 Paint GS • Amnesia • Arctic Fox • Award Maker Plus • Bard's Tale II • Betterworking Word Processor • Beyond Castle Wolfenstein • Black Magic • Bookends Extended • Bop & Wrestle • Chess 7.0 • Chessmaster 2000 • Deluxe Paint GS • Destroyer • Hacker II • Hacker II GS • Hardball • Infiltrator • Instant Music GS • J-Bird • Mabel's Mansion • Marble Madness • Mean 18 GS Golf • Megabots • Might & Magic • Miner 2049er II • Mouse Word • Music Construction Set GS • Music Studio GS • New Oregon Trail • Paintworks Plus 1.0 GS • Paintworks Plus 1.01 GS • Paul Whitehead Teaches Chess • PHM Pegasus • Poetry Express • Print Shop color version • Rambo: First Blood part II • Rocky Horror Show • Sargon III\* • Shanghai GS • Spindizzy • TelePorter • Temple Of Apshai trilogy • Top Draw GS • Transylvania • Ultima I • World's Greatest Baseball Game

**47** *Features:* Infocom-text Reader Enhancement • Color Ultimapper mod to Ultimapper IV • Towne Mapper utility for Ultima IV • Dungeon Mapper utility for Bard's Tale • Hardware Corner: Interrupting Your Apple • Softkey for Charlie Brown's 1.2.3s • *RDEX Softkeys:* • Guitar Wizard • Gemstone Warrior • Notable Phantom • Micro Wine Companion • Stickybear Printer • Note Card Maker • Starcross • Wishbringer • Dinosaur Dig • Dam Busters • Pirate Adventure • Infiltrator • MECC software • Banner Catch • Turtle Tracks • PFS File • Microzine #12, #13, #14 • Marble Madness • Writer Rabbit • Arcticfox • Age Of Adventure • Might And Magic • Space Station • Alternate Reality • Mindshadow • Gemstone Warrior • Strip Poker • Lucifer's Realm • Manuscript Manager • Bank Street Writer III • Kids On Keys • The Missing Ring • Graphic Solution • Empire I, II • Championship Golf

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

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

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

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

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

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

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

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

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

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

# Available Back Issues

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

**20** *Softkeys* • Sargon III • Wizardry: Proving Grounds of the Mad Overlord and Knight of Diamonds • The Report Card V1.1 • Kidwriter • *Feature* • Apple 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** *Softkeys* • Rendezvous With Rama • Peachtree's Back To Basics Accounting System • HSD Statistics Series • Arithmetickle • Arithmekicks and Early Games for Children • *Features* • Double Your ROM Space • Towards a Better F8 ROM • The Nibbler: A Utility Program to Examine Raw Nibbles From Disk • *Core* • The Games of 1984: In Review-part II

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

**1** *Softkeys* • Data Reporter • Multiplan • Zork • *Features* • PARMs for Copy II Plus • No More Bugs • APT's for Choplifter & Cannonball Blitz • 'Copycard' Reviews • Replay • Crackshot • Snapshot • Wildcard

# the COMPUTIST

Software Package	II +	II gs	Software Package	II +	II gs
816 Paint.....	\$45.00	<input type="checkbox"/>	\$45.00	<input type="checkbox"/>	
Airheart.....	\$24.00	<input type="checkbox"/>			
Award Maker Plus.....	\$24.00	<input type="checkbox"/>			
Bank Street Writer 64K.....	\$46.00	<input type="checkbox"/>			
Bank Street Speller.....	\$46.00	<input type="checkbox"/>			
Bank Street Mailer 64K.....	\$46.00	<input type="checkbox"/>			
Bank Street Filer 64K.....	\$46.00	<input type="checkbox"/>			
Bank Street Writer Plus (128K).....	\$46.00	<input type="checkbox"/>			
Bank Street Mailer 128K.....	\$46.00	<input type="checkbox"/>			
Bank Street Filer 128K.....	\$46.00	<input type="checkbox"/>			
Beagle Graphics.....	\$35.00	<input type="checkbox"/>			
Beyond Zork.....	\$27.00	<input type="checkbox"/>			
Black Cauldron.....	\$27.00	<input type="checkbox"/>			
Captain Goodnight.....	\$24.00	<input type="checkbox"/>			
Championship Karate.....	\$14.00	<input type="checkbox"/>			
Clip Art Gallery (for Paintworks Plus).....	\$20.00	<input type="checkbox"/>			
Copy II Plus.....	\$23.00	<input type="checkbox"/>			
Dazzle Draw.....	\$40.00	<input type="checkbox"/>			
Defender of the Crown.....	\$32.00	<input type="checkbox"/>			
Destroyer.....	\$25.00	<input type="checkbox"/>	\$27.00	<input type="checkbox"/>	
DiskQuirk.....	\$18.00	<input type="checkbox"/>			
Draw Plus.....	\$52.00	<input type="checkbox"/>			
F-15 Strike Eagle.....	\$24.00	<input type="checkbox"/>			
Fantavision.....	\$34.00	<input type="checkbox"/>	\$40.00	<input type="checkbox"/>	
Flight Simulator II.....	\$36.00	<input type="checkbox"/>			
Gamemaker.....	\$34.00	<input type="checkbox"/>			
GPLE.....	\$29.50	<input type="checkbox"/>			
The Graphics Studio.....	\$42.00	<input type="checkbox"/>			
Gunship.....	\$27.00	<input type="checkbox"/>			
Hacker II.....	\$25.00	<input type="checkbox"/>	\$30.00	<input type="checkbox"/>	
Hardball.....	\$22.00	<input type="checkbox"/>	\$28.00	<input type="checkbox"/>	
Hitchhiker's Guide.....	\$20.00	<input type="checkbox"/>			
Infiltrator II.....	\$20.00	<input type="checkbox"/>			
Jet.....	\$29.00	<input type="checkbox"/>			
Karateka.....	\$24.00	<input type="checkbox"/>			
King's Quest II.....	\$32.00	<input type="checkbox"/>	\$32.00	<input type="checkbox"/>	
Knight of Diamonds.....	\$23.00	<input type="checkbox"/>			
Labrinth.....	\$20.00	<input type="checkbox"/>			
Legacy of Lylgamyn.....	\$26.00	<input type="checkbox"/>			
Loderunner.....	\$24.00	<input type="checkbox"/>			
Macroworks.....	\$20.00	<input type="checkbox"/>			
Math Blaster.....	\$28.00	<input type="checkbox"/>			
Math Rabbit.....	\$27.00	<input type="checkbox"/>			
Mean 18, Ultimate Golf.....	\$28.00	<input type="checkbox"/>			
Merlin ProDOS.....	\$66.00	<input type="checkbox"/>			
Might & Magic.....	\$32.00	<input type="checkbox"/>			
Millionaire II.....	\$33.00	<input type="checkbox"/>			
Mousewrite.....	\$99.00	<input type="checkbox"/>	\$99.00	<input type="checkbox"/>	
Multiscribe.....	\$45.00	<input type="checkbox"/>	\$66.00	<input type="checkbox"/>	
Music Studio.....	\$52.00	<input type="checkbox"/>			
Paintworks Plus.....	\$52.00	<input type="checkbox"/>			
Portal.....	\$30.00	<input type="checkbox"/>			
The Print Shop.....	\$34.00	<input type="checkbox"/>	\$40.00	<input type="checkbox"/>	
Print Shop Companion.....	\$27.00	<input type="checkbox"/>			
Print Shop Graphics Library:					
Disk One.....	\$16.50	<input type="checkbox"/>			
Disk Two.....	\$16.50	<input type="checkbox"/>			
Disk Three.....	\$16.50	<input type="checkbox"/>			
Pro-Byter.....	\$20.00	<input type="checkbox"/>			
Racter.....	\$30.00	<input type="checkbox"/>			
Reader Rabbit.....	\$27.00	<input type="checkbox"/>	\$34.00	<input type="checkbox"/>	
Rescue Raiders.....	\$23.00	<input type="checkbox"/>			
Rocky's Boots.....	\$34.00	<input type="checkbox"/>			
Science Tool Kit: Master Module.....	\$46.00	<input type="checkbox"/>			
SDI.....	\$32.00	<input type="checkbox"/>			
Shanghai.....	\$24.00	<input type="checkbox"/>	\$30.00	<input type="checkbox"/>	
Silent Service.....	\$23.00	<input type="checkbox"/>			
Solo Flight.....	\$23.00	<input type="checkbox"/>			
Space Quest.....	\$32.00	<input type="checkbox"/>	\$32.00	<input type="checkbox"/>	
Star Trek- The Kobayashi Alternative.....	\$27.00	<input type="checkbox"/>			
Star Trek II- The Promethean Prophecy.....	\$27.00	<input type="checkbox"/>			
Stickybear Series:					
ABC's.....	\$24.00	<input type="checkbox"/>			
Math I.....	\$24.00	<input type="checkbox"/>			
Math II.....	\$24.00	<input type="checkbox"/>			
Numbers.....	\$24.00	<input type="checkbox"/>			
Reading.....	\$24.00	<input type="checkbox"/>			
Shapes.....	\$24.00	<input type="checkbox"/>			
Typing.....	\$24.00	<input type="checkbox"/>			
Writer.....	\$24.00	<input type="checkbox"/>			
Sub Battle Simulator.....	\$25.00	<input type="checkbox"/>	\$27.00	<input type="checkbox"/>	
Sub Mission.....	\$27.00	<input type="checkbox"/>			
Summer Games.....	\$14.00	<input type="checkbox"/>			
Super Macroworks.....	\$30.00	<input type="checkbox"/>			
Tass Times in Tonetown.....	\$24.00	<input type="checkbox"/>	\$24.00	<input type="checkbox"/>	
The Tower of Myraglen.....	\$32.00	<input type="checkbox"/>			
Thexder.....	\$23.00	<input type="checkbox"/>			
Topdraw.....	\$66.00	<input type="checkbox"/>			
Type.....	\$30.00	<input type="checkbox"/>			
Typing Tutor IV.....	\$32.00	<input type="checkbox"/>	\$32.00	<input type="checkbox"/>	
Video Vegas.....	\$18.00	<input type="checkbox"/>			
Visualizer.....	\$53.00	<input type="checkbox"/>	\$59.00	<input type="checkbox"/>	
Where in USA is Carmen San Diego.....	\$26.50	<input type="checkbox"/>			
Where in World is Carmen San Diego.....	\$30.00	<input type="checkbox"/>			
Winter Games.....	\$25.00	<input type="checkbox"/>	\$27.00	<input type="checkbox"/>	
Wizardy.....	\$32.00	<input type="checkbox"/>			
Word Attack.....	\$28.00	<input type="checkbox"/>			
Wordperfect w/ Spelling Checker.....	\$95.00	<input type="checkbox"/>	\$95.00	<input type="checkbox"/>	
World Games.....	\$25.00	<input type="checkbox"/>	\$27.00	<input type="checkbox"/>	
Writer's Choice Elite.....	\$60.00	<input type="checkbox"/>			
Writer Rabbit.....	\$24.00	<input type="checkbox"/>			
Zork Trilogy.....	\$46.00	<input type="checkbox"/>			

Name _____	ID# _____	● Cost of Software
Address _____		SUBTOTAL _____
City _____ State _____ Zip _____		● Plus SHIPPING _____
Country _____ Phone _____		● Plus (in WA) Sales TAX _____
  _____	Exp. _____	
Signature _____	CP50	■ TOTAL _____

# Shopper

Do Your  
Holiday Shopping  
Now

Check our  
**LOW**  
software prices  
for Apple II &  
IIGs software

## How To Order

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

Send orders to:

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

## APPLE COMPATIBLES LOWEST PRICES ANYWHERE!

1MEG/80 COL BD w/256K+Software (Ile) .....	\$99
1MEG RAM BD for IIGS w/256K .....	\$89
Above w/512K add \$39. 1 MEG add \$99	
64K/80 COL BD. New Lower Power (Ile) .....	\$35
16K RAM Board (II+) .....	\$35
128K RAM BD. New Lower Power (II+) .....	\$69
80 Column Board. Videx Comp. (II+) .....	\$49
Super Serial Board (II+/e) .....	\$49
Graphic Par BD w/6FT CBL (II+/e/gs) .....	\$45
Z80 CP/M BD Microsoft Comp. (+/e/gs) .....	\$38
Cooling Fan w/surge protect (II+/e) .....	\$29
GS Super Cooling Fan (IIGs) .....	\$25
Numeric Keypad. 16 Keys (Ile) .....	\$35
Joystick (Specify II+/c/e/gs) .....	\$15
Joystick w/Fire on stick (+/c/e/gs) .....	\$25
Mini Vacuum Cleaner w/ Attachments .....	\$10
A/B Switchbox. Parallel or RS-232 .....	\$29
Disk Drive H/H (Specify II+/c/e/gs) .....	\$129
Disk Controller Board (II+/e/gs) .....	\$39
Eprom Programmer (II+/e/gs) .....	\$49

ONE YEAR WARRANTY ON ALL PRODUCTS

CALL/WRITE FOR COMPLETE LIST  
ADD \$3 SHIPPING (Per ORDER, not per item)

**NEXO DISTRIBUTION**

914 East 8th Street, Suite 109

National City, CA 92050

(619) 474-3328 10am-6pm Mon-Fri

UNIV & SCHOOL P.O.'s WELCOME!  
VISA/MC OKAY—C.O.D. ADD \$2.00

## A Hackers Challenge

awaits those who enjoy "breaking" protection schemes for fun.

### Knowledgeable with DOS 3.3 and associated utilities?

If so, this game is for you. Includes graphics routines from **The Graphics Magician®** by Polarware™.

Inside, protected by layers of graphical puzzles and DOS-tricks *hides a secret*. Being among the 1st ten to discover the *game secret* wins you a computer-game prize! What are these prizes? I'm not saying, yet all the *clues* you need to win along with the *prize-list* are concealed in the game.

*Find the clues by playing the game.*

*Find the clues by using your favorite utilities.*

To answer this challenge send \$10.00 to:

**Mark V. Whitehurst**  
PO Box 485  
Franklin Park, IL 60131

Requires APPLE IIe or IIc.

# COMPUTIST BACK ISSUES ORDER FORM

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

Issue	Mag \$4.75	Disk \$9.95	Both \$12.95
49	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
48	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
47	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
46	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
45	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
44	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
43	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
42	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
41	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
39	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
38	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
37	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
36	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
35	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
34	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
33	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
32	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
31	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
29	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
28	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
27	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
★ 23	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
22	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
★ 7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Core 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Core 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Computing 3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Best of Hardcore Computing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Core Special Combo \$10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<small>(All three CORE magazines)</small>			

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

★ We have a limited supply of these issues.  
● Back issue is no longer available

## COMPUTIST

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

### Back Issues and Library Disk Rates

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

### What is a library disk?

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

- Library disks are available for all issues of COMPUTIST # 1 thru 45. A description of the softkeys and programs covered in each issue is available upon request. Please send your name and address along with a first class postage stamp (US \$ .22).

### Complete Your Collection!

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

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

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

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

Please send the back issues indicated:

Please send the following library disks:

Name \_\_\_\_\_ ID# \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Country \_\_\_\_\_ Phone \_\_\_\_\_

  \_\_\_\_\_ Exp. \_\_\_\_\_

Signature \_\_\_\_\_ CP50

Send check or money order to:

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

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

US funds drawn on US banks

How to get your letter published in  
**COMPUTIST's...**

**Readers Data EXchange**

### No 'Input' or 'Readers Softkey & Copy Exchange'

The old INPUT and the READERS SOFTKEY AND COPY EXCHANGE columns have now been combined into the new Readers Data Exchange (or RDEX, pronounced AR-DEX).

All softkeys, comments, articles, questions, answers, fixes and bugs and almost any other information that must be printed in a timely fashion will be placed in RDEX.

Although we will print letters and article-softkey submissions that are received as only printed text, we would prefer that they be submitted on disk.

### We no longer purchase articles or softkeys

That's right. The author's 'by-line' is the only payment offered to COMPUTIST writers who get published. Even the staff is composed of unpaid volunteers.

Of course, COMPUTIST reserves the right to edit and publish ANY contribution (whether submitted as a letter-to-the-editor or article-softkey) in the RDEX section.

All published contributions become the property (and are copyrighted as part) of COMPUTIST magazine.

### How to get your name on RDEX contributors list

Whether you are contributing data to RDEX or just asking a question, we urge you to follow these writer's guidelines:

- All contributions and submissions should be submitted in **both** printed format and on disk.
- Your printout must use only one side of the paper.
- Text should be double-spaced using 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 first page 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.

### A standard text file on a DOS 3.3 5¼" floppy disk

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.

We always try to return your disks as soon as possible. Still, to protect ourselves, we must say that we are not responsible for unreturned submissions.

### Here's how best to 'talk' to COMPUTIST readers

**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 and good, readable style.

**B.** 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 manufacturer's or publisher's name when a commercial program or product is mentioned.

**D.** When submitting your own 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. A program will not be 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** must contain detailed step-by-step procedures. Number each step

For each softkey, first introduce the locking technique used and then give precise steps to unlock the copy-protected program.

### Please do NOT send any original commercial disks

We accept articles which explain locking techniques used in several programs published by the same company.

**H. APTs** or **ADVANCED PLAYING TECHNIQUES** 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.

Please mail all letters, short articles and softkeys to the *RDEX Editor*. When accepted for publication, they will be **IMMEDIATELY** edited, typeset and entered into the next RDEX.

Remember that we cannot purchase these contributions.

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

# The Hacker's Ultimate Copy & Deprotection Utility

# Super IOB Collection !

ALL of our Super IOB controllers (through 1986) in 'ONE' package!

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

### The Super IOB program.

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

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

### This package contains:

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

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

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

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

### The SUPER IOB Collection

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

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

**Volume 2** of the Super IOB collection covers all the controllers from COMPUTIST No. 27 through No. 36. The following 65 controllers are on volume 2:

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

### ■ Yes, please send me The Super IOB Collection

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

- US/Canada/Mexico for \$16.00
- Other Foreign for \$20.00

Send to: **Super IOB Collection**  
PO Box 110846-T Tacoma, WA 98411

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

\*Requires at least 64K of memory.

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

Name \_\_\_\_\_ ID# \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Country \_\_\_\_\_ Phone \_\_\_\_\_

  \_\_\_\_\_ Exp. \_\_\_\_\_

Signature \_\_\_\_\_ CP50

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

US funds drawn on US banks