

For The Serious User Of Apple ][ Computers

# COMPUTIST

Issue No. 47

September 1987

USA \$3.75

Canada / Mexico \$7.00

All Others \$13.25

## ● Features:

Kevin Sartorelli's *Enhancement For... Infocom-Text Reader*

Dave Ciotti's *Color Ultimapper mod to Ultimapper IV*

Greg Bernhardt's *Towne Mapper utility for Ultima IV*

George Bigelow's *Dungeon Mapper utility for Bard's Tale*

Jeff Drelling's *Softkey for... Charlie Brown's 1:, 2:, 3 s*

## ● RDEX Softkeys for:

*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*

*Strip Poker*

*Lucifer's Realm*

*Manuscript Manager*

*Bank Street Writer III*

*Kids On Keys*

*The Missing Ring*

*Graphic Solution*

*Empire I, II*

*Championship Golf*

NEW

READERS  
DATA  
EXCHANGE

COMPUTIST  
PO Box 110846-T  
Tacoma, WA 98411

BULK RATE  
U.S. Postage

PAID

Tacoma, WA  
Permit No. 269

# Coping With COMPUTIST

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

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

**■ What Is A Softkey?** 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 softkeyed, the resulting disk can usually be copied with Apple's COPYA program (on the DOS 3.3 System Master Disk).

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

## ↓ PR#6

The **RETURN** key must be pressed at the end of every such command unless otherwise specified. No need to type the "]" prompt.

Control characters are specially boxed:

\* **6** **Ⓜ**

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

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

## ■ Software Recommendations:

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

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

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

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

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

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

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

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

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

*Apple II Plus - Apple IIe - Apple compatibles:*

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

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

*Apple IIe - Apple IIc:* 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 IIc warranty.

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

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

↓ **10HOME:REMCLEAR SCREEN**

The LIST will look like:

↓ **10 HOME : REM CLEAR SCREEN**

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

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

↓ **CALL -151**

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

When finished, return to BASIC with:

\* **E003G**

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

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

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

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

## ■ CHECKSOFT Instructions:

**LOAD filename  
BRUN CHECKSOFT**

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

## ■ CHECKBIN Instructions:

↓ **CALL -151**

\* **BLOAD program filename**

Install CHECKBIN at an out of the way place

\* **BRUN CHECKBIN, A\$6000**

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

\* **xxx.xxx** **Y**

Correct the lines at which the checksums differ.



## You have a LEGAL RIGHT to an unlocked backup copy

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

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

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

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

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

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

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



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

# Can't Find COMPUTIST Anywhere?

## Stop searching and subscribe now!

### annual subscription rates (12 issues):

- U.S. Subscriptions - sent third class - **\$32**
- U.S./Canada/Mexico - sent First Class - **\$45**
- U.S./Canada/Mexico First Class PLUS library disk - **\$100**
- All other Foreign Subscriptions - **\$75**
- All other Foreign PLUS library disk - **\$140**

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

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

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

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

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

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

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

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



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

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

Address \_\_\_\_\_

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

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

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

Signature \_\_\_\_\_ CP47

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

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

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





# the **COMPUTIST** shopper

Animate.....	\$43.00	Lode Runner.....	\$22.00	Sargon III.....	\$25.00
Award Maker.....	\$23.75	<b>Magic Mindow II</b> .....	<b>\$88.00</b>	Silent Service.....	\$23.00
Bank Street Writer + (128K).....	\$46.75	<b>Magic Window //e</b> .....	<b>\$88.00</b>	Summer Games II.....	\$25.00
Black Cauldron.....	\$25.00	Math Blaster.....	\$27.00	<b>Super Macroworks</b> .....	<b>\$29.50</b>
Certificate Maker.....	\$28.00	Math Rabbit.....	\$25.00	Where in the USA Carmen San Diego.....	\$26.50
<b>Copy II+</b> .....	<b>\$23.00</b>	Might & Magic.....	\$32.00	Where in the World Carmen San Diego.....	\$30.00
F15 Strike Eagle.....	\$23.00	Multiplan.....	\$62.00	Wizardry.....	\$32.00
Flight Simulator II.....	\$35.00	Music Studio (GS).....	\$52.00	<b>Word Perfect w/ Spelling Checker</b> .....	<b>\$95.00</b>
<b>GPLe</b> .....	<b>\$29.50</b>	Newsroom.....	\$35.00	WordPerfect (GS).....	\$95.00
Gamemaker.....	\$32.00	Paintworks + (GS).....	\$52.00	World Games.....	\$25.00
Hacker II.....	\$25.00	<b>Printshop</b> .....	<b>\$32.00</b>	Writer Rabbit.....	\$25.00
Hacker II (GS).....	\$30.00	<b>Printshop Companion</b> .....	<b>\$26.00</b>	Writers Choice Elite (GS).....	\$60.00
Hitchhiker Guide.....	\$20.00	Reader Rabbit.....	\$25.00	Zork I.....	\$25.00
Karateka.....	\$22.00	Reader Rabbit (GS).....	\$32.00	Zork Trilogy.....	\$45.00
Legacy of Lylgamyn.....	\$25.00	Rocky's Boots.....	\$32.00		

## How To Order

- **US orders:** Circle your selection. If total order is less than \$200, please add \$2 per item shipping & handling. Orders over \$200 receive **free** shipping. Most orders shipped UPS, so please use street address.
- In Washington state, please add 7.8% sales tax.
- Offer good while supplies last. All products are for the Apple II unless otherwise specified.
- **Foreign Orders:** Please inquire as to appropriate shipping fees.

Name \_\_\_\_\_ ID# \_\_\_\_\_  
 Address \_\_\_\_\_  
 City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
 Country \_\_\_\_\_ Phone \_\_\_\_\_  
  Exp. \_\_\_\_\_  
 Signature \_\_\_\_\_ CP47

Send orders to: **SoftKey Publishing PO Box 110816-T Tacoma, WA 98411 (206) 474-5750**

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

# The Book of Softkeys

## Volume I: Compiled from issues 1-5

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

## Volume II: Compiled from issues 6-10



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

## Volume III: Compiled from issues 11-15

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

## To order: The Book of Softkeys

- Volume I - \$7.95 + \$2 shipping/handling**  
 **Volume II - \$12.95 + \$2 shipping/handling**  
 **Volume III - \$17.95 + \$2 shipping/handling**  
 **All three volumes! - \$30.00 + \$2 ship/handling**

Name \_\_\_\_\_  
 Address \_\_\_\_\_  
 City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_  
 Country \_\_\_\_\_ Phone \_\_\_\_\_  
  Exp. \_\_\_\_\_  
 Signature \_\_\_\_\_ CP47

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

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



## RDEX Contributors

in order of appearance...

1. John Lockard
2. Greg Poulos
3. Brian Symonds
4. Michael Davis
5. David Goforth
6. Franco Panizzon
7. Bryan Sunda
- 8, 10. Dudley Brett
9. Bryan Ferrel
11. George Cawthorne
13. Mario Chrysanthou
- 14, 30. Eric W. Taylor
15. Donnie Cambre
15. Johny Garret
- 16, 33. Charles Taylor
17. M. Ferreira
18. Bob Hazelwood
- 19, 29. Jim S. Hart
20. Ronald E. Keeton
21. Frank Villaroman
22. Leigh Rowan-Kelley
23. Wayne Williams
24. Dr. Dracks
25. Dennis Gaunt
26. Brian A. Troha
27. Michael Cook
28. Jerrel Watts
31. Marc Batchelor
32. Christopher James Dean
35. Mike Newliep

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 at our standard rates of payment. SoftKey publishing purchases all and exclusive rights. 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 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, Canada & Mexico \$45, Foreign \$75. Direct inquiries to: COMPUTIST, Subscription Department, PO Box 110846-T, Tacoma, WA 98411.

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

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

## Features:

### 7 An Enhancement For... Infocom-Text Reader

by Kevin Sartorelli

This patch to G & j Bigelow's Infocom-Text Reader program makes the printed sectors more readable by translating the abbreviations. Zork I is used as an example.

### 8 Color Ultimapper mod to Ultimapper IV

by Dave Ciotti

An addition to Ultimapper IV that permits the printing of color maps. This update to the popular Ultima-utility eliminates tedious hand-coloring.

### 10 Towne Mapper utility for Ultima IV

by Greg Bernhardt

Finally, you can print a map of Britannia. Be lost no more, adventurer, as you explore the realms of Ultima IV.

### 12 Dungeon Mapper utility for Bard's Tale

by George Bigelow

Lost in the darkness of the Tower again? It's time to make a map that can lead you safely through the dungeons.

### 15 Softkey for... Charlie Brown's 1, 2, 3 s

by Jeff Dressing

### 18 Put Your Bit-Copy Utilities Into Fast-Access RAM Drive

by Jerry D. Greer

### 41 Hardware Corner: How To Interrupt Your Apple

by Bobby

## 21 Readers Data Exchange

Softkeys: software title (RDEX #)

Guitar Wizard—2 • Gemstone Warrior—3 • Notable Phantom—3 • Micro Wine Companion—3 • Stickybear Printer—3 • Note Card Maker—4 • Starcross—5 • Wishbringer—5 • Dinosaur Dig—5 • Dam Busters—5 • Pirate Adventure—6 • Infiltrator—7 • MECC software—8 • Banner Catch—10 • Turtle Tracks—10 • PFS File—10 • Microzine #12, #13, #14—10 • Marble Madness—12 • Writer Rabbit—14 • Arcticfox—15 • Age Of Adventure—16 • Infiltrator—17 • Might And Magic—18 • Space Station—19 • Alternate Reality—22 • Mindshadow—23 • Gemstone Warrior—25 • Strip Poker—26 • Lucifer's Realm—26 • Manuscript Manager—26 • Bank Street Writer III—30 • Kids On Keys—31 • The Missing Ring—31 • Graphic Solution—32 • Empire I, II—33 • Championship Golf—35 •

N:notes APT: A:adventure Tips BUG:

APT:Wizardry—1 • BUG:Taking the Wiz out of Wizardry—1 • BUG:softkey for Mousepaint—5 • N:Hex-Decimal Conversion Tables—5 • APT:Wizardry—9 • N:How To Make Microsparc's Assembler and ProDOS Macrosoft run on a Laser 128—19 • N:Using ARD on RDOS—24 • N:Add FastDOS to Swordthrust—26 • N:Unlocking //gs software—26 • N:Improving Locksmith's Fast Backup Routine—26 • A:Castle Wolfenstein's SS men—27 • N:Exodus Construction Set—28 • APT:Lost Tomb unlimited whips and men—29 • N:Fool the Graphics Expander! Convert Newsroom Clip Art to Print Shop Graphics Expander disks—34 •

## Attention (volunteer) staff.

### Thank-you

...for making it possible for me to continue publishing COMPUTIST in spite of diminishing subscriptions and spiraling expenses that threatened to spell the demise of this 6-year old publication.

I have taken into consideration all of your suggestions and have agreed to most of them:

### More Information Per Issue

Yes, the purpose of COMPUTIST was to provide a forum and media of data exchange especially for information about removing copy-protection from locked up commercial software. And I agree that the previous editorial policies prevented the timely release of that information. The policy of verifying each softkey contributed to this delay and I will not require it any more. As was brought out in earlier discussions, our contributors and readers do their homework well and if a submitted softkey doesn't work for everyone, then other readers will discover this and send in a note or even new and better softkeys.

### Readers Data Exchange

The INPUT column and READERS SOFTKEY AND COPY EXCHANGE is now consolidated. I agree that sometimes it was difficult to distinguish the two when the INPUT contained softkeys, too. This new section that you have called The Readers Data Exchange is, however, on a trial basis and MUST prove that it delivers more information to COMPUTIST subscribers as fast and as soon as you say.

### Where's The Cover

Lowering expenses has been difficult and personally painful as I see COMPUTIST going from a slick, stiff-covered magazine to a self-covered newsletter and will tolerate it only until publishing debts are erased and I can again put COMPUTIST in a glossy cover.

### From 32 to 48 plus pages

In Issue 47, I have decided to increase the number of pages from 32 to 48. I feel that it is only fitting that the loss of the cover, which resulted in a very large saving, be offset by at least an increase of pages, which I expect to be filled with still more Readers Data Exchange.

### Readers are our Writers

As you have suggested, cost-cutting will include payments to the writers as well. For all the writers with whom I have published material, Payment will be made. Letters will be going out to other writers that will explain your new editorial policy and payment procedures. I feel we may lose some talented hackers and I hope you are right in your assertion that most readers-writers are loyal to COMPUTIST and will understand that these drastic cost-cutting measures are necessary to the survival of this magazine.

### Well Done

Considering the circumstances, you've done a professional job and I hope that you will keep to the deadlines I made.

I understand that all of you have other commitments and I thank you for the time you've spent on COMPUTIST.

Please notify our readers that the technical help-line is no longer offered.

I will continue to find ways to increase subscriber levels while decreasing expenses so that we may someday return to a paid-staff profitability.

*Charles R. Haight  
Publisher*



# Infocom-Text Reader

by Kevin Sartorelli


**■ Requirement:**

- *Infocom-Text Reader* program (from George & John Bigelow's article: *Infocom Revealed*, COMPUTIST 34 page 16)

I would like to present an enhancement to the excellent article by George and John Bigelow in COMPUTIST No. 34 ('Infocom Revealed'). This enhancement will not affect the way the program works, only the printout it can produce.

In order to use the enhancement you will have to decode the abbreviations for the disk you wish to work with. The method for doing this is explained in the original article. I found that to decode the abbreviations you have to list them all out as you think they go and then check your list against some phrases on the disk. I normally found that I had missed a few and had to go back and find them.

When your list of abbreviations is complete add them as data statements to the end of the enhanced reader program (see the examples with this article). When you now go to print out a sector the abbreviations are expanded to make the text much more readable.

Note that with the enhancement installed you must have at least one abbreviation or the program will crash in line 155. Also your interface card must not echo to the screen (the  80N in line 640 turns off the echo for most printer cards).

You will find that the printout is slower with the enhancement and you may wish to keep the original program around for initial investigations of a new disk.

I hope this patch makes the printed sectors much more readable.

## Patch To Infocom-Text Reader

```

155 READ N: DIM A$(N), B$(N): FOR I = 1 TO N
: READ A$(I), B$(I): NEXT: REM
: READ ABBREVIATIONS
610 VTAB 24: HTAB 7: PRINT
"DOUBLESPACE?(Y/N)": : GET A$
620 IF A$ <> "Y" AND A$ <> "y" AND A$ <
> "N" AND A$ <> "n" THEN VTAB 22:
PRINT: GOTO 610
630 VTAB 22: PRINT: VTAB 24: PRINT
"WORKING...": : VTAB 1
640 PR# 1: PRINT CHR$(9) "80N"
    
```

```

650 FOR Y = 2 TO 22
660 FOR X = 1 TO 40
670 VTAB Y: HTAB X
680 XY = SCRN(X-1, 2*(Y-1)) + 16 *
: SCRN(X-1, 2*(Y-1) + 1)
690 B$ = B$ + CHR$(XY - 128)
700 NEXT X
710 FOR I = 1 TO LEN(B$): D$ = MID$(B$, I, 2)
: K = 0
720 IF PEEK(-16368) > 128 THEN C$ = "" : B$
= "" : POKE -16368, 0: GOTO 800
730 IF LEFT$(D$, 1) > "@" THEN D$ = LEFT$
(D$, 1): GOTO 760
740 FOR J = 1 TO N: IF D$ = A$(J) THEN K = J: J
= N: I = I + 1
750 NEXT J: D$ = LEFT$(D$, 1): IF K <> 0
THEN D$ = B$(K) + " "
760 C$ = C$ + D$: NEXT I
770 VTAB 1: HTAB 1: PRINT C$
780 IF A$ = "Y" OR A$ = "y" THEN PRINT
790 B$ = "" : C$ = "" : NEXT
800 VTAB 1: PRINT
810 PR# 0
820 VTAB 24: HTAB 1: PRINT "PRESS A KEY
TO CONTINUE-": : GET A$
830 GOTO 170
840 REM -- ABBREVIATIONS --
850 DATA 1: REM NUMBER OF ABBREVIATIONS
KNOWN
860 DATA "DUD", "DUD"
    
```

### Checksums

155	- \$8F74	740	- \$AD8B
610	- \$2926	750	- \$E993
620	- \$27C7	760	- \$8834
630	- \$D5E6	770	- \$0D6D
640	- \$DF3E	780	- \$B649
650	- \$93DC	790	- \$4D4F
660	- \$0570	800	- \$9791
670	- \$201B	810	- \$4E50
680	- \$62A9	820	- \$8E83
690	- \$7046	830	- \$4958
700	- \$5385	840	- \$E5B3
710	- \$5A2B	850	- \$7EDE
720	- \$1A92	860	- \$40F0
730	- \$8928		

### Sample Abbreviation List For Zork I

```

840 REM ABBREVIATIONS FOR ZORK I
850 DATA 115: REM NUMBER OF ABBREVIATIONS
KNOWN
860 DATA "<A", "THE", "<<<", "?THE", "<=",
"?YOU", "<<<", "?", "<?", "AND"
870 DATA "<@", "YOUR", "<A", "IS", "<B",
"YOU", "<C", "?THERE", "<D", "OF"
880 DATA "<E", "CAN'T", "<F", "TO", "<G",
",", "<H", "WITH", "<I", "ARE"
890 DATA "<J", "THAT", "<K", "WHICH", "<L",
"?THIS", "<M", "HAVE", "<N", "FROM"
900 DATA "<O", "LARGE", "<P", "APPEARS",
"<Q", "THROUGH", "<R", "IN", "<S",
"HERE"
    
```

```

910 DATA "<T", "ROOM", "<U", "WATER",
"<V", "CANNOT", "<W", "?CYCLOPS",
"<X", "NARROW"
920 DATA "<Y", "SMALL", "<Z", "INTO",
"=A", "LEADING", "=<", "SOUTH", "=",
"CYCLOPS"
930 DATA "=>", "HIS", "=?", "THIEF", "=@",
"SEEMS", "=A", "BE", "=B", "BUT"
940 DATA "=C", "?IT", "=D", "THIS", "=E",
"ALREADY", "=F", "THERE", "=G", "NORTH"
950 DATA "=H", "?YOUR", "=I", "WEST", "=J",
"EAST", "=K", "CAN", "=L", "DON'T"
960 DATA "=M", "NOT", "=N", "ON", "=O",
"STAIRCASE", "=P", "PROBABLY", "=Q",
"FOR"
970 DATA "=R", "WOULD", "=S", "TROLL", "=T",
"HERE", "=U", "LOOKING", "=V",
"CANDLES"
980 DATA "=W", "PATH", "=X", "DOOR", "=Y",
"?ROOM", "=Z", "ABOUT", ">A", "ROOM"
990 DATA "><", "NOTHING", ">=",
"?FRIGID", ">>", "?UNDERGROUND",
">?", "GROUND", ">@", "?A"
1000 DATA ">A", "?I", ">B", "WALL", ">C",
"PASSAGES", ">D", "IMPOSSIBLE", ">E",
"OUT"
1010 DATA ">F", "HAS", ">G", "LEVEL",
">H", "CLOSED", ">I", "WON'T", ">J",
"KNOCKS"
1020 DATA ">K", "PASSAGE", ">L",
"?FORTUNATELY", ">M", "IT", ">N",
"TOO", ">O", "OTHER"
1030 DATA ">P", "TROLL'S", ">Q",
"DOESN'T", ">R", "WOODEN", ">S",
"VERY", ">T", "BEEN"
1040 DATA ">U", "RAINBOW", ">V",
"GRATING", ">W", "AN", ">X", "THINK",
">Y", "RIVER"
1050 DATA ">Z", "COULD"
1060 DATA "@C", "0", "@D", "1", "@E", "2",
"@F", "3", "@G", "4", "@H", "5", "@I",
"6", "@J", "7", "@K", "8", "@L", "9"
1070 DATA "@M", " ", "@N", " ", "@O", "!",
"@P", "?", "@S", " ", "@W", "-", "@X",
":", "@Y", "(", "@Z", ")"
    
```

### Checksums

840	- \$B57B	960	- \$0254
850	- \$E569	970	- \$7D28
860	- \$55F1	980	- \$38EB
870	- \$0A71	990	- \$BB08
880	- \$266B	1000	- \$8D67
890	- \$C62C	1010	- \$EFC5
900	- \$6B9D	1020	- \$D4DC
910	- \$A9FB	1030	- \$02CF
920	- \$6812	1040	- \$F33C
930	- \$4494	1050	- \$5324
940	- \$2180	1060	- \$B797
950	- \$7418	1070	- \$25E5



A graphic utility for...

# Ultima IV

# Color

a  
Colorful  
Update  
to  
Ultimapper  
IV

by Dave Ciotti

■ Requirements:

- Apple II
- Various colored ribbons for your printer
- Ultima IV*

*Ultimapper IV* (see COMPUTIST 33) is a great help in playing *Ultima IV*, the well-known adventure game.

### Hand-color the map?

But when I started to hand-color the map, after printing it, I decided it was smarter to let my printer do the coloring. If you like the idea, here's how I did it:

First, because a color version requires at least five printer passes, it makes sense to speed up the original black and white program.

Lines 470, 480, and 490 check to see if the value on the disk map is valid and then print the appropriate character. A question mark is printed if the value on the disk is unknown. By making V(n) an ASCII look-up table with 63's, the ASCII value of question mark in unknown locations.

Lines 470 - 490 of the original program can be reduced to a single PRINT statement.

Two new arrays must be dimensioned. I used CO\$(5) for colors and CO(5,7) for the characters to be printed in each color.

Data statements and a For-Next loop are needed to fill the new arrays.

We also need a prompt to change ribbons, the logic to print only that desired color, and a loop to print all the colors.

My essential changes to the *Ultimapper IV* program perform these function.

In addition, I also changed the first two DATA statements, to make the map more understandable, but these alterations are not absolutely necessary.

### A Bug squashed

Furthermore, the original third line of DATA contained a bug. I have corrected it.

Finally, I added five more DATA lines to the program.

The binary file SECTOR.ZAP is still used.

### Typing it in

Use the instructions on the inside front cover of this magazine, to key in the BASIC portion of the program and save it with:

**I SAVE COLOR ULTIMAPPER IV**

Next, key in the hexdump and save it to the same disk with:

**I BSAVE SECTOR.ZAP,A\$300,L\$21**

### Printing Problems In Color

To print multiple colors on the same sheet of paper, you need an accurate way to position the paper for each color.

The method I use is to tape together (in a loop) two more sheets of paper than I actually need for the printing (eighteen in this case).

Set the top of form just before printing the first color.

When printing is completed, three form-feeds will position the paper correctly for the next color.

**NOTE:** Change ribbons only when the print head is over one of the unused pieces of paper.

This program is very slow. It takes almost ten hours to print all five colors (blue, green, red, yellow, and black).

However, if you can download characters to your printer, you can further speed up the program by using again the technique I employed earlier to eliminate the print color check!

### Sector.Zap Hexdump

0300:	A9 03 A0 0A 20 D9 03 60	\$6D5D
0308:	00 00 01 60 01 00 00 00	\$37AD
0310:	1B 03 00 95 00 00 01 00	\$08D4
0318:	00 60 01 00 01 EF D8 00	\$C3C5
0320:	00	\$4343



# Ultimapper

## Color Ultimapper

```

10 REM *-----*
20 REM *   ULTIMA IV   *
30 REM * MAINLAND PRINTER *
40 REM *   BY JIM S. HART *
50 REM *-----*
60 REM
70 REM *-----*
80 REM * SET UP VARIABLES *
90 REM *-----*
100 REM
110 HIMEM: 38144

120 IF PEEK (768) < > 169 OR PEEK (769) <
  > 3 THEN PRINT : PRINT CHR$ (4) : "BLOOD
  SECTOR. ZAP, A$300"
130 CLEAR : BUF = 38143 : RWTS = 768 : SEC = 0 :
  COLOR = 0 : DIM LOC (16, 256) : V (255)
  : V$ (27) : CO (5, 7)
140 HOME : D$ = CHR$ (4) : P0$ = D$ + "PR#0"
  : P1$ = D$ + "PR#1"
150 FOR I = 0 TO 255 : V (I) = 63 : NEXT : FOR I =
  1 TO 27 : READ V, J : V (V) = J : NEXT : IV$
  = "ULTIMA IV"
160 FOR X = 1 TO 5 : READ CO$ (X) : FOR Y = 1 TO
  7 : READ CO (X, Y) : NEXT : NEXT
170 REM
180 REM *-----*
190 REM * PRINT OUT TITLE SCR N *
200 REM *-----*
210 REM
220 HOME : INVERSE
230 VTAB 10 : PRINT "ULTIMA IV MAINLAND
  PRINTER"
240 NORMAL : VTAB 20
250 PRINT "INSERT BRITANNIA & PRESS
  RETURN"
260 PRINT "(" : SPC (9) : "ESC TO EXIT" :
  SPC (9) : ")" :
270 GET AS : ON (AS < > CHR$ (13) AND AS <
  > CHR$ (27)) GOTO 270
280 PRINT AS : IF AS = CHR$ (27) THEN TEXT :
  HOME : END
290 HOME : VTAB 10 : INVERSE

```

```

300 CO = CO + 1 : PRINT "INSERT" CO$ (CO)
  CHR$ (7) "RIBBON, POSITION PAPER,"
  : HTAB 11 : PRINT "AND HIT
  RETURN" : INPUT "" : AS
310 NORMAL
320 REM
330 REM *-----*
340 REM * TURN ON PRINTER AND *
350 REM * PRINT OUT MAINLAND *
360 REM *-----*
370 REM
380 PRINT P1$ : PRINT CHR$ (9) "80N"
390 PRINT IV$
400 POKE 790, 1 : POKE 791, 0
410 FOR LOOP = 0 TO 15
420 FOR S1 = SEC TO (SEC + 3)
430 POKE 782, LOOP : POKE 783, S1 : POKE 791
  : 0 : CALL RWTS : POKE 72, 0
440 FOR J = 1 TO 256 : LOC (S1, J) = PEEK (BUF +
  J) : NEXT
450 NEXT
460 FOR I = 0 TO 15
470 FOR S1 = SEC TO (SEC + 3)
480 FOR J = 1 TO 16
490 Y = (V (LOC (S1, J + (16 * I))) ) ) : FOR X
  = 1 TO 7 : IF Y = CO (CO, X) THEN PRINT
  CHR$ (Y) : X = 7 : NEXT : GOTO 510
500 NEXT : PRINT "A" :
510 NEXT
520 NEXT : PRINT "A"
530 NEXT
540 NEXT
550 IF SEC < > 12 THEN PRINT CHR$ (12) : SEC
  = SEC + 4 : GOTO 390
560 TEXT : HOME : PRINT P0$ : SEC = 0 : IF CO <
  5 THEN 290
570 END
580 REM *-----*
590 REM * VALUE OF ITEM & *
600 REM * CHR REPRESENTATION *
610 REM *-----*
620 REM
630 DATA 0, 115, 1, 126, 2, 94, 3, 35, 4, 33, 5
  , 45, 6, 43, 7, 47, 8, 42
640 DATA 9, 68, 10, 99, 11, 67, 12, 86, 13, 91
  , 14, 64, 15, 93
650 DATA 23, 61, 24, 37, 29, 77, 30, 83, 55
  , 124, 60, 36, 76, 38

```

```

660 DATA 68, 49, 69, 50, 70, 51, 71, 52
670 DATA BLUE, 115, 126, 94, 50, 63, 256, 256
680 DATA GREEN, 35, 33, 45, 43, 49, 256, 256
690 DATA RED, 68, 99, 67, 86, 77, 83, 52
700 DATA YELLOW, 61, 37, 36, 51, 38, 256, 256
710 DATA BLACK, 47, 42, 124, 91, 64, 93, 256

```

## Checksums

10	- \$BADD	370	- \$3FD5
20	- \$9B13	380	- \$528C
30	- \$4D3B	390	- \$336C
40	- \$AD92	400	- \$B712
50	- \$C899	410	- \$75EA
60	- \$FF65	420	- \$5426
70	- \$A3BF	430	- \$DB39
80	- \$A900	440	- \$E032
90	- \$924D	450	- \$72A1
100	- \$CB63	460	- \$AC7B
110	- \$B755	470	- \$05C6
120	- \$4038	480	- \$E1F9
130	- \$74C5	490	- \$8FC6
140	- \$4FB8	500	- \$50A1
150	- \$CF41	510	- \$DD91
160	- \$C2D6	520	- \$0187
170	- \$84BE	530	- \$5C65
180	- \$0403	540	- \$E693
190	- \$65E8	550	- \$91B8
200	- \$F20C	560	- \$2D7B
210	- \$5636	570	- \$9244
220	- \$54F8	580	- \$6323
230	- \$3602	590	- \$38C9
240	- \$9723	600	- \$FA0D
250	- \$3860	610	- \$D626
260	- \$5959	620	- \$09F7
270	- \$0E92	630	- \$C9A1
280	- \$F9CA	640	- \$04D2
290	- \$A377	650	- \$9895
300	- \$C474	660	- \$8337
310	- \$C304	670	- \$438B
320	- \$BCBF	680	- \$ED59
330	- \$3B5B	690	- \$AF1E
340	- \$0F3E	700	- \$6287
350	- \$670E	710	- \$C29B
360	- \$1EDB		

Another graphic utility for...

# Ultima IV

# Towne

by Greg Bernhardt

### ■ Requirements:

- Apple II plus or better
- Printer
- Ultima IV*

As most *Ultima IV* players are well aware of, Britannia is a big place! Thanks to Jim S. Hart's article in *COMPUTIST* 33, I was able to find my way around the mainland, but I was still lost whenever I entered a town, village or castle.

This prompted me to create the program presented here.

0300	A9 03 A0 09 20 D9 03 60	\$5A40
0308	00 01 60 01 00 00 00 1A	\$1407
0310	03 00 40 00 00 01 00 00	\$2024
0318	60 01 00 01 EF D8 00	\$AFB9

### Keying It In

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

#### 1 SAVE ULTIMA TOWNE MAPPER IV

Next, use the procedure for keying in hexdumps to type in the machine language portion of this program and save it with:

#### 1 BSAVE OBJ.ULTIMA TOWNE MAPPER, A\$300, L\$1F

You are now ready to get those maps you have been anxiously awaiting.

### Using the Program

As usual, you can start up the program with:

#### 1 RUN ULTIMA TOWNE MAPPER IV

The program will load the "OBJ" file automatically, if it needs to.

Use of the program is fairly straight forward. As soon as the program starts up, insert your Towne disk in the same drive as the disk you ran the program from.

You will be presented with a list of possible maps. Use left and right arrow keys will move the inverse bar to the map you desire. When you get to the map that you want, press RETURN.

If you choose the "LIST OF SYMBOLS" option, you will be given a legend of all the symbols used when printing maps.

Have fun, and good luck in Britannia!

### Obj.Ultima Towne Mapper

0300:	A9 03 A0 09 20 D9 03 60	\$5A40
0308:	00 01 60 01 00 00 00 1A	\$1407
0310:	03 00 40 00 00 01 00 00	\$2024
0318:	60 01 00 01 EF D8 00	\$AFB9

### Ultima Towne Mapper IV

```

10 REM .....
20 REM ::ULTIMA::TOWNE::
30 REM ::::MAPPER:IV::::
40 REM .....
50 REM :BY::::
60 REM ::GREG:BERNHARDT::
70 REM .....
80 REM
90 ST = 16384 :TR = 781 :SE = 782 :IV = 1 :SL =
  1 :DIM MP$(18)

100 TEXT : HOME :A$ =
  "ULTIMA△TOWNE△MAPPER△IV" : GOSUB 770
110 A$ = "BY△△GREG△BERNHARDT" : GOSUB 770 :
  POKE 34 , 3
120 IF PEEK (768) = 169 AND PEEK (769) = 3
  THEN 140
130 PRINT CHR$( 4 )
  "BLOAD△OBJ.ULTIMA△TOWNE△MAPPER"
140 VTAB 12 :A$ = "INSERT△TOWNE△DISK△NOW!"
  : GOSUB 770
150 RESTORE : FOR A = 1 TO 18 : READ MP$( A ) :
  NEXT
160 LNS$ = "" : FOR A = 1 TO 129 : READ Y :LNS$ =
  LNS$ + CHR$( Y ) : NEXT
170 REM
180 REM .....
190 REM ::GET:A:SELECTION::
200 REM .....
210 REM
220 HOME : PRINT "WHICH△WOULD△YOU△LIKE△
  TO△PRINT?" : VTAB 6
230 FOR A = 1 TO 18 : PRINT MP$( A ) : NEXT
240 VTAB IV + 5 : INVERSE : PRINT MP$( IV )
250 NORMAL : WAIT - ST , 128 : GET A$
260 IF A$ = CHR$( 13 ) THEN 380
270 IF A$ = CHR$( 27 ) THEN TEXT : HOME : END
280 IF A$ < > CHR$( 8 ) AND A$ < > CHR$( 21
  ) THEN 250
290 VTAB IV + 5 : PRINT MP$( IV )
300 IF A$ = CHR$( 8 ) THEN IV = IV - 1 : IF IV =
  0 THEN IV = 18
310 IF A$ = CHR$( 21 ) THEN IV = IV + 1 : IF IV =
  19 THEN IV = 1
320 GOTO 240
330 REM
  
```



# Mapper IV

```

340 REM .....
350 REM ::PRINT:A:TOWNE::
360 REM .....
370 REM
380 IF IV = 18 THEN 560
390 HOME : PRINT "CREATE△MAP△OF△"
    MP$(IV) " " : PRINT "(ESCA=△QUIT)"
400 PRINT : PRINT "LOADING△MAP,△HOLD△ON."
410 PRINT CHR$(4) "BLOAD△MAP" CHR$(63 +
    IV) " ,A$4000"
420 GOSUB 680
430 FOR Y = 0 TO 31
440 SP = ST + (Y * 32)
450 FOR Z = 0 TO 31
460 MM = PEEK (SP + Z)
470 PRINT MID$(LN$,MM + 2,1);
480 IF PEEK (- ST) = 155 THEN POKE - 16368 , 0
    : GOTO 500
490 NEXT : PRINT : NEXT
500 PR# 0 : CALL 1002 : GOTO 220
510 REM
520 REM .....
530 REM ::SYMBOLS::
540 REM .....
550 REM
560 HOME : PRINT "CREATE△LIST△OF△SYMBOLS:"
    : PRINT "(ESCA=△QUIT)"
570 GOSUB 680
580 PRINT "LIST△OF△SYMBOLS△FOR△ULTIMA△
    TOWNE△MAPPER△IV:"
590 PRINT : PRINT
600 FOR A = 1 TO 35 : READ X , X$
610 PRINT "△" CHR$(X) "△=△" X$ : NEXT
620 PR# 0 : CALL 1002 : GOTO 150
630 REM
640 REM .....
650 REM ::TURN:ON:PRINTER::
660 REM .....
670 REM
680 PRINT : PRINT : PRINT "PRINTER△IS△IN△
    SLOT=>" SL : CHR$(8) : : GET A$
690 IF A$ < > CHR$(13) AND A$ < > CHR$(
    27) THEN SL = VAL (A$)
700 IF A$ = CHR$(27) THEN POP : GOTO 220
710 PRINT A$ : PR# SL : CALL 1002 : RETURN
720 REM
730 REM .....
740 REM ::CENTER:A$:
750 REM .....
760 REM

```

```

770 HTAB 20 - LEN (A$) / 2 : PRINT A$ : RETURN
780 REM
790 REM .....
800 REM ::DATA::
810 REM .....
820 REM
830 DATA BRITANIA△LEVEL△2 , BRITANIA△
    LEVEL△1 , THE△LYCACEUM , EMPATH△ABBEY
    , SERPENTS△HOLD , MOONGLOW , BRITAIN
    , JHELOM , YEW
840 DATA MINOC , TRINSIC , SKARA△BRAE
    , MAGINCIA , PAWS , BUCCANEERS△DEN
    , VESPER , COVE , LIST△OF△SYMBOLS
850 DATA 88 , 32 , 32 , 126 , 59 , 58 , 94 , 42 , 83
    , 36 , 79 , 37 , 38 , 37 , 38 , 61
860 DATA 38 , 115 , 115 , 115 , 115 , 104 , 104 , 79
    , 61 , 98 , 61 , 61 , 94 , 118 , 37 , 110
870 DATA 89 , 77 , 77 , 84 , 84 , 70 , 70 , 68 , 68
    , 73 , 73 , 82 , 82 , 80 , 80 , 83
880 DATA 83 , 48 , 92 , 47 , 47 , 92 , 79 , 43 , 82
    , 47 , 82 , 73 , 73 , 123 , 124 , 45
890 DATA 45 , 39 , 39 , 39 , 39 , 33 , 33 , 33 , 33
    , 35 , 64 , 117 , 40 , 34 , 46 , 111
900 DATA 111 , 88 , 88 , 88 , 88 , 88 , 88 , 88 , 88
    , 88 , 88 , 88 , 88 , 88 , 88
910 DATA 88 , 65 , 66 , 67 , 68 , 69 , 70 , 71 , 72
    , 73 , 74 , 75 , 76 , 77 , 78 , 79
920 DATA 80 , 81 , 82 , 83 , 84 , 85 , 86 , 87 , 88
    , 89 , 90 , 61 , 93 , 91 , 63 , 32 , 35
930 DATA 32 , DEEP△OR△REGULAR△WATER , 126
    , SHALLOW△WATER , 59 , BOG , 58 , GRASSLAND
940 DATA 94 , BRUSH , 42 , FOREST , 83 , HILLS , 36
    , MOUNTAINS , 79 , DUNGEON
950 DATA 37 , TOWN ;△RUIN△OR△VILLIAGE , 38
    , CASTLE , 115 , SHIP , 104 , HORSE , 79
    , HEX△FLOOR
960 DATA 61 , BRIDGE , 98 , BALLOON , 61
    , TOP△OR△BOTTOM△BRIDGE , 94 , LADDER△UP
970 DATA 118 , LADDER△DOWN , 110 , SHRINE , 48
    , PILLIAR , 92 , SLEEPING△PERSON , 82
    , ROCKS
980 DATA 73 , DOOR△LOCKED , 73 , DOOR△UNLOCKED
    , 123 , CHEST , 124 , ANKH , 45 , FLOOR
990 DATA 33 , FORCE△FIELDS , 35 , WALL , 34
    , LAVA , 93 , RIGHT△TABLE , 91 , LEFT△TABLE
1000 DATA 61 , MIDDLE△TABLE , 63 , WINDOW

```

## checksums

10	- \$BADD	510	- \$2473
20	- \$9B13	520	- \$089D
30	- \$4D3B	530	- \$56E4
40	- \$AD92	540	- \$032B
50	- \$C899	550	- \$B0DD
60	- \$FF65	560	- \$4A6A
70	- \$A3BF	570	- \$6DAE
80	- \$A900	580	- \$9DA8
90	- \$DBEC	590	- \$92F4
100	- \$1E98	600	- \$E180
110	- \$1F04	610	- \$2876
120	- \$4165	620	- \$2545
130	- \$94FC	630	- \$1637
140	- \$091C	640	- \$ECBB
150	- \$E33E	650	- \$A8A0
160	- \$0671	660	- \$87F1
170	- \$9F24	670	- \$C2D1
180	- \$2E40	680	- \$FF62
190	- \$7AEC	690	- \$16A2
200	- \$1E67	700	- \$26E9
210	- \$F1BC	710	- \$79AF
220	- \$B740	720	- \$2CD0
230	- \$97AF	730	- \$2F3E
240	- \$6AE0	740	- \$998E
250	- \$D4F6	750	- \$7191
260	- \$F2FB	760	- \$0AC0
270	- \$2D7B	770	- \$4153
280	- \$6AB5	780	- \$9227
290	- \$F447	790	- \$C669
300	- \$13B9	800	- \$C886
310	- \$997E	810	- \$C37F
320	- \$93C0	820	- \$FE99
330	- \$C036	830	- \$4360
340	- \$0EF6	840	- \$694F
350	- \$9691	850	- \$0924
360	- \$7500	860	- \$9141
370	- \$4004	870	- \$396D
380	- \$E770	880	- \$0125
390	- \$CB50	890	- \$ADCB
400	- \$CCD2	900	- \$128D
410	- \$E580	910	- \$192C
420	- \$2E91	920	- \$8AA1
430	- \$6F96	930	- \$3E57
440	- \$1F75	940	- \$F58D
450	- \$5616	950	- \$8009
460	- \$3771	960	- \$AD38
470	- \$33A9	970	- \$68DF
480	- \$77AC	980	- \$36E1
490	- \$5600	990	- \$6BB9
500	- \$509C	1000	- \$BC57

A graphic utility for...

# BARD'S Tale

# Dungeon

by George Bigelow

■ Requirements:

- The Bard's Tale
- A hi-res screen-to-printer dump program

*The Bard's Tale* is a great game, engrossing and varied, but is it ever tough! Some great aids for the game have been published recently, but something that would help immensely would be maps for the dungeons.

Ever since I got totally lost in the dark in the Tower I realized that a mapper was a necessity!

So I got out the sector editor and, to make a long story short, here it is.

But first there are some questions that you should ask.

## What is a Room?

Each dungeon is 22 rooms wide by 22 rooms long. Each room is represented by a hex value. Each hex value indicates where a door or wall is to be drawn.

For instance, 05 is a wall to the north and a wall to the south (used in drawing an east to west corridor).

The strange thing is that each room is defined as that which you see from the inside of that room; from the outside some walls or doors may not be there! This means that you can walk into a room, turn around, and there's a wall!

What this boils down to is that each room must match its neighbor; if a room has a door to the west, the next room must have a door to the east.

## Passages may be 1 or 2-way

This rule is not strictly followed by the *Bard's Tale* designers, in some places you can walk through a door, turn around, and it's a wall.

There are 131 hex values that are used to represent rooms. To make things a lot easier I have opted to not show doors, which drops the number of room types to draw to 16. So a dungeon map drawn by this program will only show ways that are open to travel, secret doors or not.

One thing, though, if a room has a door to the east, and the next room has a wall to the west, the map will show a wall there. So if you come upon a door that the map shows as a wall, if you go through the door and turn around there will be a wall.

## Why All The Data Statements?

The data statements with all the zeros in them save a lot of time in the mapping, and lots of your typing.

The alternative is for you to type in 16 IF-THEN statements containing all the 131 room values.

So as you are typing in all those numbers be glad it's so few!

Each of those values is a room type to be HPLOTted for that room value. Why 231 and not 131? Not all the room values are used.

## Two Maps Per Room?

Yes, oddly enough. Looking at the disk you see four consecutive sectors; two with lots of hex numbers, and two that are mostly zeros and eights.

The first two, properly processed, form the maze and the second two are what I call the Effects Map.



# Mapper

This map shows the stairs, monsters, areas of darkness, traps, etc. To help you find where on the map the effect is I have made the Effects Map overlay the Hi-Res map with dots wherever the program finds a non-zero value.

## How Do I Use The Effects Map?

First print out the Hi-Res Map, using your favorite hi-res screen dump program.

Then print the Effects Map. It corresponds directly to the Hi-Res map in that wherever there is a dot there is a non-zero value in the Effects Map.

These values (there are only 16 of them) mean the following:

<i>value</i>	<i>object</i>
<i>1, 2, 5, and 9</i> .....	<i>stairs.</i>
<i>80 and 88</i> .....	<i>monsters.</i>
<i>8 and 88</i> .....	<i>darkness.</i>
<i>10, 18, 1C, and 90</i> .....	<i>traps.</i>
<i>0C and 84</i> .....	<i>inscriptions.</i>
<i>20</i> .....	<i>holes in the floor.</i>
<i>40</i> .....	<i>holes in the ceiling.</i>
<i>4 . is the most interesting. This is the catch-all number. It can stand for descriptions, words, a throne, special warriors, direction shifts, special items, and I don't know what all else.</i>	

All these numbers are approximate; I haven't spent much time on them. If you want accuracy, it's up to you!

So, back to using the Effects Map. Write on your Hi-Res map whatever it is that you figure out from the Effects Map.

Sorry about all that work, but the alternative is to create shape tables, 16 of them, and I want to get back to playing the game!

## Why The Hex Map?

Well, first it was very easy to include, and second it'll be handy if you want to punch a hole through a wall somewhere with your sector editor (00 is a hole, or no walls).

The sector starts in the southeast corner of the dungeon and makes a row to the east. The 23rd byte then starts the next row to the north of the previous one.

And so on...

## What If I Don't Have a Printer?

You can still draw the Hi-Res map and copy it from the screen, but the Hex Map and the Effects Map do need a printer (unless you have an 80 column screen. It might work then. I don't, so don't quote me).

## How Does It Work?

The chosen sector (and most of the preceding one) are read into the R array.

Going through the array by rows (the last 22 bytes, then the next to last 22 bytes, etc.) we

use the value obtained from the R array to get us a value from the T array (the long one, remember?). This value gets us to the right HPLLOT statement and the room is drawn.

Then the same method is used to overlay the effects dots.

Same method again to print the other two maps, only poking the decimal value into the very short machine language routine at 776 that will print out the hex value. Simple in theory...

## Any Suggestions?

Well, yes. Head for all the fours on your Hi-Res map. Mark them on the map as you come across them. Most will be signs over doors, direction shifters, smoke in the eyes, and so forth but hidden among them are the clues you need and some interesting items.

For instance, the southeasternmost 4 in CASTLE(7) is a crystal sword.

## Hints and Question

Also head for the 0C and 84s. These are hints and questions you need to answer to finish the game.

If you wish a larger map do this: get rid of line 20, change N in line 30 to 8, and change HGR in line 460 to HGR2.

If you wish a smaller map, just make N smaller.

This program won't win the game for you (you'll still have to interpret clues, answer questions, and fight monsters) but at least now you won't have to spend so much time in the dungeons and you can be sure you have visited everywhere worthwhile in them. Good luck!

# Bard's Tale Dungeon Mapper

```

100 REM BARD'S TALE
200 REM DUNGEON MAPPER
300 REM BY GEORGE BIGELOW
400 REM
500 REM COPR. 1987 COMPUTIST
600 REM
700 LOMEM: 16384
800 DIM R(484) : DIM M$(15) : DIM T(255) : N =
  7 : P = INT (N / 2)
900 FOR A = 768 TO 781 : READ B : POKE A, B :
  NEXT : REM RWTS & HEX PRINT ROUTINES
1000 FOR A = 0 TO 15 : READ M$(A) : NEXT : REM
  LOCATIONS
1100 FOR A = 0 TO 231 : READ T(A) : NEXT : REM
  H PLOT ROUTINE TO USE
1200 DATA 169,183,160,232,32,217,3,96
  ,169,0,32,218,253,96A
1300 DATA ΔCELLARS(0),SEWERS(1),SEWERS(2)
  ,SEWERS(3),CATACOMBS(4)
  ,CATACOMBS(5),CATACOMBS(6)
1400 DATA CASTLE(7),CASTLE(8),CASTLE(9)
  ,TOWER(A),THEΔTOWER(B),THEΔTOWER(C)
  ,THEΔTOWER(D),THEΔTOWER(E)
  ,THEΔTOWER(F)
1500 DATA 0,1,0,0,3,6,3,3,0,1,0,0,0
  ,1,0,0,4,7,4,4,10,13,10,10,4,7
  ,4,0,4,7,0,0,0,1,0,0,3,6,3,3
  ,0,1,0,0,0,0,0,0,1,0,0,3,6
  ,3
1600 DATA 0,0,1,0,0,0,0,0,2,5,2,2
  ,8,11,8,8,2,5,2,0,2,5,0,0,9,14
  ,9,9,12,15,12,12,9,14,9,9,9,14
  ,9,0,2,5,2,0,8,11,8,0,2,5,2,0
  ,0,0
1700 DATA ΔΔ2,0,2,0,0,0,8,11,8,0,0
  ,5,2,0,0,0,0,0,0,1,0,0,3,6,3
  ,0,0,0,0,0,0,0,0,0,4,7,4,4,10
  ,13,10,0,4,7,4,4,0,0,0,0,0,1,0
1800 DATA 0,4,6,4,0,0,1,0,0,0,0,0,0
  ,0,0,0,0,0,0,0,0,0,0,0,0,0,0
  ,0,0,0,1,0,0,3,6,0,0,0,0,0,0
  ,0,0,0,0,4,7,4,0,0,13,10,0,0,7
1900 DATA 0,0,0,0,0,0,0,0,0,0,6,0
  ,3
2000 REM MENU
2100 HOME : PRINT "CHOOSEΔΔΔLOCATION:" :
  PRINT
2200 FOR A = 0 TO 15 : PRINT A + 1 ". " SPC(1 + (A
  < 9))M$(A) : NEXT
2300 PRINT "17.ΔEND"
2400 POKE - 16368,0 : REM CLEAR ANY PREVIOUS
  CHARACTER
2500 PRINT : INPUT "CHOOSEΔΔΔNUMBERΔ(1-17)
  :Δ" : C : IF C < 1 OR C > 17 THEN 210
2600 IF C = 17 THEN END
2700 REM CALCULATE TRACK & SECTOR
2800 T = 15 - INT (C / 2 + .5) : D = C - 1
2900 IF C = 1 THEN S = 15 : GOTO 320
3000 IF C = 2 THEN S = 7 : GOTO 320
3100 C = C - 2 : GOTO 290
3200 POKE 47083,0 : POKE 47084,T
3300 HOME : FLASH : PRINT "INSERTΔBARD'SΔ
  TALEΔDUNGEONΔDISK!" : NORMAL
3400 PRINT : PRINT "LOCATIONΔCHOSEN:Δ" :
  INVERSE : PRINT M$(D) : NORMAL
3500 PRINT : PRINT "CHOOSEΔTYPEΔOFΔMAP:"
3600 PRINT "1.ΔHI-RESΔMAP"
3700 PRINT "2.ΔEFFECTSΔMAPΔ(TURNΔONΔ
  PRINTER)"
3800 PRINT "3.ΔHEXΔMAPΔ(TURNΔONΔPRINTER)"
3900 PRINT "4.ΔRETURNΔTOΔMAINΔMENU"

```

```

400 PRINT : PRINT "CHOOSEΔΔΔNUMBERΔ(1-4) :
  Δ" : GET Z : PRINT Z
410 ON Z GOTO 440,860,860,210
420 GOTO 330
430 REM LOAD IN 1st SECTOR & PUT INTO ARRAY
440 GOSUB 970
450 REM H PLOT THE ARRAY
460 HGR : HCOLOR=3 : X=0 : Y=0
470 HOME : VTAB 22 : PRINT "MAPΔOFΔ" M$(D) :
480 FOR ROW = 22 TO 1 STEP - 1
490 FOR C = 22 * ROW - 21 TO 22 * ROW
500 IF R(C) > 231 THEN 540
510 AA = T(R(C)) + 1
520 ON AA GOTO 540,550,560,570,580,590
  ,600,610,620,630,640,650,660,670
  ,680,690
530 REM H PLOT ROUTINES
540 GOTO 700 : REM 0
550 H PLOT X, Y TO X + N, Y : GOTO 700 : REM 1
560 H PLOT X, Y TO X, Y + N : GOTO 700 : REM 2
570 H PLOT X, Y + N TO X + N, Y + N : GOTO 700 :
  REM 3
580 H PLOT X + N, Y TO X + N, Y + N : GOTO 700 :
  REM 4
590 H PLOT X + N, Y TO X, Y TO X, Y + N : GOTO 700
  : REM 1,2
600 H PLOT X, Y TO X + N, Y : H PLOT X, Y + N TO X
  + N, Y + N : GOTO 700 : REM 1,3
610 H PLOT X, Y TO X + N, Y TO X + N, Y + N : GOTO
  700 : REM 1,4
620 H PLOT X, Y TO X, Y + N TO X + N, Y + N : GOTO
  700 : REM 2,3
630 H PLOT X, Y TO X, Y + N : H PLOT X + N, Y TO X
  + N, Y + N : GOTO 700 : REM 2,4
640 H PLOT X, Y + N TO X + N, Y + N TO X + N, Y :
  GOTO 700 : REM 3,4
650 H PLOT X + N, Y TO X, Y TO X, Y + N TO X + N, Y
  + N : GOTO 700 : REM 1,2,3
660 H PLOT X, Y TO X, Y + N TO X + N, Y + N TO X +
  N, Y : GOTO 700 : REM 2,3,4
670 H PLOT X, Y TO X + N, Y TO X + N, Y + N TO X, Y
  + N : GOTO 700 : REM 1,3,4
680 H PLOT X, Y + N TO X, Y TO X + N, Y TO X + N, Y
  + N : GOTO 700 : REM 1,2,4
690 H PLOT X, Y TO X + N, Y TO X + N, Y + N TO X, Y
  + N TO X, Y : REM 1,2,3,4
700 X = X + N : NEXT C
710 Y = Y + N : X = 0 : NEXT ROW
720 REM OVERLAY EFFECTS DOTS
730 S = S - 2 : GOSUB 970
740 X = 0 : Y = 0
750 FOR ROW = 22 TO 1 STEP - 1
760 FOR C = 22 * ROW - 21 TO 22 * ROW
770 IF R(C) < > 0 THEN H PLOT X + P, Y + P
780 X = X + N : NEXT C
790 Y = Y + N : X = 0 : NEXT ROW
800 PRINT CHR$(7) : PRINT CHR$(7)
810 POKE - 16368,0
820 HOME : VTAB 21 : PRINT "PRESSΔQΔTOΔ
  QUIT,ΔANYΔOTHERΔKEYΔTOΔRETURNΔTOΔ
  MENU" :
830 GET QS : IF QS = "Q" THEN HOME : VTAB 22 :
  PRINT "MAPΔOFΔ" M$(D) : END
840 TEXT : GOTO 210
850 REM EFFECTS OR HEX MAPPER
860 S = S - 2 * (Z = 2) : GOSUB 970
870 PRINT : PRINT CHR$(4) "PR#1"
880 FOR ROW = 22 TO 1 STEP - 1
890 FOR C = 22 * ROW - 21 TO 22 * ROW
900 POKE 777,R(C) : CALL 776
910 PRINT "Δ" :
920 NEXT : PRINT : NEXT
930 PRINT : PRINT "MAPΔOFΔ" M$(D)
940 PRINT : PRINT CHR$(4) "PR#0"
950 GOTO 210

```

```

960 REM READ SECTORS INTO R ARRAY
970 POKE 47085,S : CALL 768
980 FOR A = 1 TO 256 : R(A) = PEEK (A + 38399)
  : NEXT
990 POKE 47085,S - 1 : CALL 768
1000 FOR A = 1 TO 228 : R(A + 256) = PEEK (A +
  38399) : NEXT
1010 RETURN

```

## Checksums

10	- \$BADD	520	- \$044D
20	- \$9B13	530	- \$A678
30	- \$4D3B	540	- \$6F22
40	- \$AD92	550	- \$ABFB
50	- \$C899	560	- \$4A45
60	- \$FF65	570	- \$B6FA
70	- \$19E4	580	- \$E389
80	- \$007C	590	- \$5CDA
90	- \$6050	600	- \$8E79
100	- \$F342	610	- \$B3B7
110	- \$CE86	620	- \$67EF
120	- \$0EAB	630	- \$9CCC
130	- \$3EBD	640	- \$17D2
140	- \$A923	650	- \$1479
150	- \$C985	660	- \$4FFD
160	- \$DDE8	670	- \$0BAF
170	- \$1FAF	680	- \$C7E0
180	- \$8315	690	- \$BD47
190	- \$A026	700	- \$7E76
200	- \$2C41	710	- \$1B00
210	- \$55B7	720	- \$2F64
220	- \$2996	730	- \$EE87
230	- \$8558	740	- \$2226
240	- \$1C3C	750	- \$CBF7
250	- \$AF21	760	- \$F359
260	- \$1EF3	770	- \$B37F
270	- \$4743	780	- \$3122
280	- \$F8E3	790	- \$3D56
290	- \$6D18	800	- \$E270
300	- \$4476	810	- \$168C
310	- \$FD42	820	- \$93CD
320	- \$06BF	830	- \$06AD
330	- \$BAC3	840	- \$B413
340	- \$46A1	850	- \$0215
350	- \$0AE9	860	- \$CD3E
360	- \$CDA0	870	- \$317D
370	- \$E4D4	880	- \$2470
380	- \$CAF2	890	- \$FC92
390	- \$3560	900	- \$0550
400	- \$EECF	910	- \$92EB
410	- \$AD63	920	- \$F050
420	- \$A730	930	- \$F5F8
430	- \$C088	940	- \$F5AD
440	- \$4119	950	- \$D8B1
450	- \$CD4A	960	- \$00E0
460	- \$7DD1	970	- \$F6C9
470	- \$2C73	980	- \$73E0
480	- \$6FE9	990	- \$D193
490	- \$926D	1000	- \$7947
500	- \$C139	1010	- \$EDD5
510	- \$B33B	0	- \$0000



## Softkey For...

# CHARLIE BROWN'S 1, 2, 3 S

Random House

by Joe Drelling

### ■ Requirements:

- Apple II Plus and up
- A DOS 3.3 disk with no "HELLO" program
- Four blank disks
- Super IOB 1.5
- Charlie Brown's 1, 2, 3's*

### Background

*Charlie Brown's 1, 2, 3's* is an excellent educational game which helps to teach numbers and counting to young children.

The games are supposed to be designed for children from preschool through the second grade but my one-and-a-half-year-old son does enjoy playing with it (with some assistance of course!).

*Charlie Brown's 1, 2, 3's* comes on the front and back side of one disk. The front side has three options to keep your child entertained. These include: 1. Number Recognition, 2. Counting, and 3. an option which combines the first two.

Side two only has one option which combines counting practice and the pre-math skill of forming sets. The counting and number recognition games include integer numbers from one thru ten.

Though the programs can be run on a monochrome monitor, all of the games take full advantage of the Apple's color capability to help keep a child's interest.

Although the animation and sound routines

aren't the best you have ever seen on your Apple, they are entertaining.

Additionally, they are used as positive reinforcement for correct answers.

As with most software packages discussed in *COMPUTIST*, the publisher (Random House) has decided to copy protect the software.

The entire package is designed to allow young children to use it unsupervised if necessary. Considering that no backup was provided and that the package cost \$39.95, I was not excited about turning over the original disk to any youngster!

Additionally there is an extensive amount of disk I/O ongoing throughout the programs. Thus I decided to attempt to deprotect it so that we could have a working copy for everyday use.

### The Investigation

When I first booted side 1 of the original disk my disk drive did not make any unusual noises such as moving back to check a particular track for a nibble count.

By all indications the disk seemed to boot normally and I even spotted the Applesoft prompt appear during the booting process. This indicated to me that Random House had probably used only a slightly modified version of Apple DOS to protect its software.

Since I didn't feel like attempting to boot code trace the disk, I decided to look for another way to deprotect the two sides.

Realizing that if I could gain access to the monitor after the software had booted I could then capture the modified Read Write Track Sector (RWTS) routines used to protect *Charlie Brown's 1, 2, 3's*. The RWTS is the portion of the Disk Operating System (DOS) which gives your computer the information it needs to be able to read information from, or write data to, the disk. Refer to *Beneath Apple DOS*

by Worth and Lechner, and your *Apple II DOS Programmer's Manual* for more information on RWTS operations. On my initial try to break through to the monitor, I booted the *Charlie Brown's 1, 2, 3's* disk and immediately pressed **ESC**.

This did not seem to have any effect so I rebooted and tried using the Escape key. The Escape key also failed to stop the booting process.

### Control-Reset

On my third try I used Control-Reset. This time I had some luck. One press of the Control and Reset keys caused the program to reboot, which wasn't what I'd hoped for so I tried Control-Reset again and met with success!

After waiting for the first screen to come up (the one with your child's name on it) I pressed and held the Control key down and then I pressed the Reset key two to three times in rapid succession. This stopped the booting process and resulted in an Applesoft prompt.

To see if I had allowed the boot time to load the modified RWTS I tried to CATALOG the disk. Typing "CATALOG" only returned a beep, indicating that DOS had been modified, so I tried the alternative way of CATALOGing a disk, **CALL 42350**. This resulted in a 34 file catalog scrolling up on my screen!

To see if Random House had utilized the same protection scheme on side two of the disk I flipped the original disk over and attempted to catalog side two by retyping "CALL 42350".

Side two of the original had been written using the same protection technique. This time I ended up with a 35 file catalog scrolling up my screen. It appeared that I would be able to capture the modified RWTS and thus be able to use the Newswrap Controller routine to deprotect the *Charlie Brown's 1, 2, 3's* disk.

But Random House decided not to let things turn out to be quite that easy.

## The Procedure

After loading the *Charlie Brown's 1, 2, 3's* version of DOS, I verified it by CATALOGing (via CALL 42350) both sides of the Charlie Brown's 1, 2, 3's disk. Then I checked to see that it would not CATALOG a standard DOS 3.3 disk. Finally I entered the monitor with:

### I CALL -151

Since the standard RWTS routine normally resides in memory between \$8B00 and \$8FFF (those are hexadecimal numbers which equate to decimal 35584 and 36863) it was necessary to move that block of memory to another area in memory which would not be affected by a warm boot (booting DOS without turning off the computer or pressing Control Open-Apple Reset at once).

That way when a warm boot was done the modified RWTS would not be overwritten during the boot.

I used the following monitor command to move the *Charlie Brown's 1, 2, 3's* RWTS to a safe area in memory:

\* 1900<8B00.8FFFF

That command moves (copies) the block of memory from \$8B00 thru \$8FFF down to lower memory starting at \$1900 (that equates to 6400 decimal).

The "M" at the end of the above command tells the monitor to "Move" the specified block of memory.

Next it was necessary to reload DOS 3.3 into memory without losing the modified RWTS. That had to be accomplished in order to save the modified RWTS, as a binary file, onto a standard DOS 3.3 disk.

This was accomplished by executing a "warm boot" from the monitor. Before the command was given for a warm boot, the original *Charlie Brown's 1, 2, 3's* disk was removed from disk drive #1 and replaced with a DOS 3.3 disk which did not have a boot program (normally called "HELLO") on it.

The warm boot was executed from the monitor by using the following command:

\* C600G

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

It is important to ensure that the DOS 3.3 disk you put into drive #1 does not have a boot program on it. If necessary temporarily rename your normal boot program to something else such as "HELLO.ORIGINAL".

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

\* BSAVE RWTS.CHARLIE BROWN'S  
123S, A\$1900, L\$800

This command stored the copy of the RWTS starting at \$1900, \$800 bytes long, into a ten sector binary file on the disk.

Once I had completed all of my preparations,

I loaded Super IOB 1.5 and entered the original Newspaper Controller from COMPUTIST No. 22, page 21.

The Newspaper Controller was updated to accept the modified RWTS as is listed below. Then I ran Super IOB 1.5 and selected the option to format the deprotected disk.

The copying process went smoothly for both sides of the *Charlie Brown's 1, 2, 3's* disk. But when I tried to boot either side of my new backup disk, it would not boot. The disk drive just kept trying to recalibrate.

Before I went any further I decided to see if the program would run if I booted a normal DOS 3.3 disk.

After I rebooted my slave disk (without the boot program) I CATALOGed the backup disk and found an Applesoft file named "HELLO". This looked like a good place to start so I ran it and the entire package on side one appeared to work properly.

The only difference I noted was that I did not see the screen which identified the primary user's (child's) name. The same was true for side two of the disk.

Since everything else worked properly after I booted a normal DOS 3.3 disk I decided to just replace the DOS tracks.

Before replacing the DOS tracks (tracks 0 thru 2) I decided to positively identify the correct boot program name.

By using a sector editor and looking at track \$1, sector \$9 I found that the actual boot program was a binary program called "STEX".

Then I used *Copy II Plus* version 5.4 to copy the DOS tracks from a fast DOS disk. By using the "Change Boot Program" option on *Copy II Plus* I changed the boot program of the newly installed DOS back to the original "STEX" file.

Then both sides of the disk booted properly and appeared to run fine.

## Further Investigations

But as I continued stepping through the programs on both sides of the disk I found that they would only work about halfway.

Then strange things started happening on my monitor and with my disk drives. So I decided that some further investigation was needed in order to identify what the problem was.

Since I knew that things were all right when I first backed up the disk I made another copy of the disk and started over.

This time I used *Copy II Plus* again to take a closer look at the copy. After looking around for a short while I found out what the problem was: Random House had stored portions (10 sectors worth) of one of its files on track 2, a DOS track. So when I copied a new DOS onto the disk I overwrote a portion of the file.

On side 1 file "SEQ9" was partially stored on track 2 and on side two it was file "SEQ1" which was stored there. Both files were stored in sectors \$5-\$8 and \$A-\$F on track \$2.

A further examination of side 1 showed that every sector on the disk was used while on side 2 there were some free sectors available on non-DOS tracks.

Now I knew that in order to copy side 1 so that it would work properly I had to identify some files that could be deleted. Then I could move the files in the DOS tracks out onto another area of the disk.

By examining the boot program "STEX" and the Applesoft program "HELLO" I was able to solve the problem. The "STEX" program doesn't do much for you other than allow you to enter your child's name (one time only), display a hi-res picture which identifies the disk as belonging to your child, and BLOAD a file which modifies DOS.

From both side one and side two I deleted two files by entering the following commands:

I DELETE STEX

I DELETE BPLATE.CPRS

I already knew that the "STEX" program wasn't really necessary because the "HELLO" program worked fine by itself. The "BPLATE.CPRS" file was the one BLOADed by "STEX" and it only modified the DOS so you couldn't use standard DOS commands (like CATALOG) when you exited the program.

Once I had identified the unnecessary programs I formatted another disk with a fast DOS and a boot program called "HELLO" by entering:

I INIT HELLO

Then I deleted the HELLO program on the newly initialized disk:

I DELETE HELLO

Finally I copied all of the files from side 1 of my deprotected disk to the newly initialized disk (except for the STEX and BPLATE.CPRS files which were just DELETED).

I then followed an identical procedure to correct the problem on side 2.

## Fixes

Once the copy process has been completed, LOAD the HELLO program from side 1 into memory and LIST it. You can DELETE line 7 which is listed below:

I 7 GOTO 7

By deleting line 7 you will be dumped back out into DOS when the program crashes. Otherwise you would end up in an infinite loop.

Next load the HELLO program from side 2 and list it. Then you can delete line 11 which is similar to line seven above. Line 11 is listed below:

I 11 GOTO 11

All of the Applesoft programs on both sides of the disk have similar lines in them. They are all statements at the end of the ONERR GOTO routine and they all go to themselves. Everyone of these statements can be safely deleted or recoded as:

I nn END

The "nn" is the appropriate line number. Normally when the ONERR routine goes into effect it is because the program had attempted



to write to the file "PARMS" and you had write-protected your disk. The names of all of the Applesoft programs and the lines which can be deleted are listed in the Table 1 below:

**Table 1:**

Lines To Delete Or Recode		
PROGRAM NAME	LINE #	
FROM SIDE 1: HELLO	7	
TITLE	74	
GAME123.V5	100	
FROM SIDE 2: HELLO	11	
TITLE	42	
GAME123.S2	91	

Before changing or deleting any of the above listed lines please ensure that they are of the form I described earlier (ie: 11 GOTO 11).

If you would like the program to check to see if the problem which caused it to crash was that your disk was write-protected then make the following change to the first line of the ONERR routine in the HELLO and TITLE programs:

Change The Following Line:

```
nn POKE 216,0: PRINT CHR$(4) "PR#0":
TEXT: HOME: PRINT CHR$(7): HTAB 10:
VTAB 7: PRINT "A disk or drive":
HTAB 10: PRINT "error has occurred.":
PRINT: HTAB 10: PRINT "Check the disk
drive": HTAB 10: PRINT "and start
again."
```

Change The Above Line To Read:

```
nn POKE 216,0: PRINT CHR$(4) "PR#0":
TEXT: HOME: PRINT CHR$(7): HTAB 10:
VTAB 7: PRINT "Disk error!": IF
PEEK(222) = 4 THEN PRINT "YOUR DISK
IS WRITE-PROTECTED!": END
```

Note that "NN" above is the appropriate line number for the program you are working on. It is the line number listed in the ONERR GOTO NN statement in line # 1 of each of the Applesoft programs.

## WARNING

Due to the large amount of memory used by the GAME123.V5 and GAME123.S2 programs do not attempt to add additional lines to either of these programs or you will lose portions of the last program line!

Though these changes are not critical to the operation of the deprotected disks, they do allow you to exit easily to DOS 3.3 without pressing the [C] keys.

If you do decide to make these changes make sure that you LOAD the program in memory, make the change in memory, DELETE the old program from disk, and then SAVE the modified program.

## Step By Step

**1** Initialize a blank disk with DOS 3.3 or a fast DOS and DELETE the boot program. Do this step only if you do not already have a DOS 3.3 disk without a boot program (normally named "HELLO").

**1** FP  
**1** INIT HELLO  
**1** DELETE HELLO

**2** Type in the below listed *Charlie Brown's 1, 2, 3's* Controller and save it as a text file called CHARLIE BROWN'S 123S CONTROLLER.

**3** Load the *Charlie Brown's 1, 2, 3's* disk (side 1) in disk drive #1 and boot it.

**1** PR#6

**4** Press and hold your Control key and then press your Reset key two to three times in rapid succession anytime after the initial screen appears.

Note that the disk may start to reboot after you first press the Reset key but will stop after you continue pressing Reset. Then the Applesoft prompt will appear on the screen.

**5** Verify that you did allow the boot sufficient time to load the modified RWTS by CATALOGing the original disk. Then enter the monitor and move the *Charlie Brown's 1, 2, 3's* RWTS.

**1** CALL 42350  
**1** CALL -151  
\* 1900<8B00.8FFF

**6** Replace the *Charlie Brown's 1, 2, 3's* disk with a DOS 3.3 disk that has no HELLO boot program. Then execute a warm boot from the monitor.

\* C600G

**7** Save the modified RWTS to disk.

**1** BSAVE RWTS.CHARLIE BROWN'S  
123S, A\$1900, L\$800

If you CATALOG your DOS 3.3 disk you should now have a ten sector binary file named RWTS.CHARLIE BROWN'S 123S.

**8** Load Super IOB 1.5 and install the controller.

**1** LOAD SUPER IOB 1.5  
**1** EXEC CHARLIE BROWN'S 123S  
CONTROLLER

**9** RUN Super IOB 1.5 and select the format disk option.

**1** RUN

**10** Load your favorite fast DOS and initialize a blank disk with a boot program

named "HELLO" and then DELETE the HELLO program.

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

**11** DELETE the two unnecessary files from your deprotected copy of this side of *Charlie Brown's 1, 2, 3's*. (Note: you must perform this step for side 1 or you will not have sufficient space available on your working disk to accept all of the necessary files!)

**1** DELETE STEX  
**1** DELETE BPLATE.CPRS

**12** Use any file transfer program such as FID to transfer all of the remaining files from the deprotected disk to the newly initialized working disk.

**13** Repeat steps 9 thru 13 above for side 2. If you desire you can use both sides of your working disk to keep the entire *Charlie Brown's 1, 2, 3's* package together on one disk.

**14** Make the recommended changes listed above to the Applesoft programs if desired. Then DELETE the old Applesoft program and resave the modified program.

**1** DELETE HELLO  
**1** SAVE HELLO

**15** Finally, let your children enjoy learning from a deprotected backup copy of *Charlie Brown's 1, 2, 3's*!

**16** If you chose not to follow step 12 for side 2 of the disk, you should rename your boot program in DOS as STEX. Then your disk will run identically to the original. If you would like to change your child's name later on look on track \$2, sector \$9 for where the name is stored. If you do change it use your sector editor and ensure that the last byte after the name is a \$00. That's all there is to it! Enjoy your deprotected copy of *Charlie Brown's 1, 2, 3's*!

## Controller

```
1000 REM CHARLIE BROWN'S 1, 2, 3'S CONTROLLER
1010 TK = 0 : LT = 35 : ST = 15 : LS = 15 : CD = WR
      : FAST = 1
1020 GOSUB 360 : GOSUB 490 : GOSUB 610
1030 GOSUB 360 : GOSUB 490 : GOSUB 610 : IF
      PEEK (TRK) = LT THEN 1050
1040 TK = PEEK (TRK) : ST = PEEK (SCT) : GOTO
      1020
1050 HOME : PRINT "COPY DONE" : END
1010 PRINT CHR$(4) "BLOAD RWTS.CHARLIEA
      BROWN'S 123S, A$1900"
```

## Checksums

1000	- \$356B	1040	- \$1078
1010	- \$2544	1050	- \$C083
1020	- \$C809	10010	- \$82FE
1030	- \$FA9D		

# Z RAM & RamWorks

# Put Your Into Fast-

by Jerry D. Greer

## ■ Requirements:

- The Applied Engineering Z-RAM or Ramworks (256K min.)
- Some favorite utility or copy programs (such as Copy II Plus)

## Applied Engineering

If you buy that Z-RAM or RamWorks card, what will you do with it? I could give you a book full of reasons to get either of these memory expansion boards.

Most owners probably will use their extra memory in association with a word processor or spreadsheet program.

However, by using the software that is supplied with the memory expansion board, you can use your utility and copy programs in more efficient ways.

My setup is based on the Apple //c and the Applied Engineering Z-RAM I with 512k of memory. Although it is primarily used with a word processor, it has proven to be a valuable addition for many other uses.

For example, I use it frequently in association with my *Copy II Plus* utility program.

They work very well together and make my work easier. Both are excellent tools and it is only natural that sooner or later they would be joined together in a Hackin' Shack.

In this article, we will look at the way the expanded memory can be used to improve your use of some utility programs.

Using *Copy II Plus* as an example, we will take a close and detailed look at a typical application.

## RAM Drive & ProDrive:

### How it speeds up access time

The software that sets up the RAM drive is called *RamDrive* for DOS 3.3 and *ProDrive* for ProDOS. Separate programs are supplied for each operating system.

Briefly, it works this way. Your memory expansion board comes with a disk of utilities that make the board very useful.

Some new programs will recognize and automatically use the extra memory now in your computer.

## Help From:

### Appleworks Super Desktop Expander

Most programs, however, require some special attention and help. *RamDrive* and *ProDrive* provide this extra help by setting up a very fast electronic disk in RAM. You can then load almost any program into the RAM disk where lightning fast access will speed up your operations.

## That Magic Utility Program:

### AUTOCOPY

The magic program to use for creating the ProDOS RAM Drive is in a file named AUTOCOPY on the *Super Appleworks Desktop Expander* utility disk supplied by Applied Engineering.

The program you need to set up a DOS 3.3 RAM Drive is located on the reverse side of the ProDOS disk.

## The Old...

The first releases of Applied Engineering's utilities were primitive. Creating the RAM Drive was easy but loading any program into the memory required a knowledge of the BASIC language.

Any of the versions of the *Appleworks Super Desktop Expander* earlier than version 5.2 required that the loader program be rewritten each time a different program was to be put into memory.

It was slow and cumbersome and was subject to the usual writing errors we all have when writing or modifying a program. If you come across a copy, it is worth looking at.



# Bit-Copy Utilities

## Access RAM 'Drive'

### Loading ProDOS Programs

The current releases of the *Expander* disk have really made loading a ProDOS program into memory very easy.

First, you select the Autocopy operation from the menu.

Next you enter the name of the volume you want to load into the RAM Drive and hit RETURN. That is all there is to it!

The AUTOCOPY program automatically loads all files from your selected program (except ProDOS in the case of ProDrive) into RAM. When using ProDOS programs, the ProDOS file is not loaded because it is already in memory.

### Loading DOS 3.3 Programs

Loading a DOS 3.3 program into the RAM Drive is nearly as easy. The instructions with the *RamDrive* program are clear and easy to follow. I had little difficulty the first time I tried it.

For Example:

### Using it with Copy II Plus 6.0

Remember that in normal use, all of the *Copy II Plus* utility is loaded anyway. If you are using only the utility, there is no advantage in loading it into a special RAM drive.

However, *Copy II Plus* basically has two separate programs on the disk. There is the utility part and there is the bit-copy part. On bootup, the utilities are loaded into memory. If you want any of the bit copy routines, you have to call them up by selecting "Bit Copy" from the copy option on the main menu.

If you work with *Copy II Plus* requires routines from both of the program areas, you will spend quite a lot of time waiting for a routine to load and reload. Having both already

loaded can save you significant time.

A look at some times that are important when using *Copy II Plus* in RAM.

### Chart 1

1. Normal utility load time.....	10 seconds
2. Bit copy load time (out of utilities).....	7 seconds
3. Utility reload time (out of bit copy).....	7 seconds
4. RAM bit copy load time.....	1 second
5. RAM utility reload time (out of bit copy).....	1 second
6. Time required for ProDrive to initially load all of the Copy II Plus Program into RAM.....	80 seconds

**Note:** The times do NOT include any time required to locate the disk and reinsert it into the drive.

### Instant Parameters

You can also save time when working with the bit copy routines and the aids that go with it.

Parameters are instantly available without lengthy search and load. This is especially true when you are using the "Try..." entries. Each is instantly available when called up.

Now that the sector editor for *Copy II Plus* is over in the bit copy routines, it is very nice to be able to call it all up in a second or two. It has always been very distracting to have ideas in mind while the computer labors slowly to bring up the tools you need to test your plans. It is a real pain to have to repeatedly insert the program disk each time you want to use something in the "other area".

Having the program completely loaded into RAM does cause some subtle but acceptable problems. For example, *Copy II Plus* knows that RAM is engaged and assumes that there is not enough memory to copy a disk in two passes.

When making disk copies, it defaults to the four tracks at-a-time copy process instead of doing it in two passes.

When going from the utility programs to the bit-copy routines, you must specify the slot number for your disk controller card. If you are using an Apple //c, the only number that you can enter is "6". All others are illegal and will not be accepted by the program. With the //e, or //+, you should enter the slot number for the actual slot where you have your disk controller card installed. After you hit return, the bit copy routine menu pops right up.

### Step by Step

How to load a ProDOS program, such as *Copy II Plus*, into RAM.

- 1 Catalog the disk to determine the operating system and volume name.
- 2 Read the catalog and note the first "System" file name in the list.
- 3 Boot the Super Appleworks Desktop Expander ver 5.2 (or later).
- 4 Select #4. "ProDrive options menu", from the main menu.
- 5 Next, select #3. "Autocopy all files to ProDrive".
- 6 Enter name of volume you want to copy and return.
- 7 Change /RAM/ subdirectory OR just press return.
- 8 Insert the program disk (*Copy II Plus*) into drive 1 and return.

**9** The autocopy routine sequentially copies files into RAM.

**10** When the program asks for the next volume name, just press RETURN.

**11** The prompt appears at screen left bottom and you can start.

## Running A Program That Is Resident In A RAM Drive

After your program is in the /RAM drive, how do you start it?

The answer to this question depends upon the kind of operating system you are using. Is it DOS 3.3 or is it ProDOS? Is it something else?

Remember that before loading any program, you have to catalog the disk to find the name of the first SYSTEM file on ProDOS disks and the HELLO program on DOS 3.3 disks.

With ProDOS (which is the *Copy II Plus 6.0* operating system) it is easy. After the Applied Engineering Autoload procedure is completed, you are given the BASIC prompt "J", and your screen is otherwise blank. Now, we cannot do a cold start or we lose the program we have carefully loaded into the memory.

The same problem occurs if we try the "open apple-control-reset" procedure.

Now, a little knowledge about ProDOS comes in handy. When you boot any ProDOS disk, the operating system will look for and run the very first "SYSTEM" file listed in the catalog.

The first SYSTEM file that we find listed in the catalog on the *Copy II Plus* disk is the "UTIL.SYSTEM." It is this file that we must first push into memory if we are to see the program come up.

ProDOS has a very handy command that really makes our task of starting up a program (that we have in memory) very easy. It is referred to as the "smart run" command. It is simply the minus symbol, the "-".

So, when we see the prompt after the Autocopy has finished its job of loading the contents of the disk, we simply type (for our example *Copy II Plus*):

```
-/RAM/UTIL.SYSTEM
```

Remember all those things that you have read about "pathnames" in ProDOS? This is where you put your knowledge to work. The command line above simply tells your Apple to find the disk named RAM and run the file UTIL.SYSTEM. When you press the RETURN key, there is a pause of a second and then *Copy II Plus* comes up!

If you are working with DOS 3.3 programs, you will use the *RamDrive* Program.

Loading a DOS 3.3 program is a little different but is basically just as simple. The *RamDrive* utility uses the old but dependable *FID* to load files from a disk into the RAM drive. After the files are copied onto the RAM drive, the *FID* menu will come back up.

Select the "Quit" option and the Applesoft prompt will appear.

We have the same restrictions with DOS 3.3 as with ProDOS. You must run the program with software commands. With DOS 3.3 you enter the command "RUN" and the name of the program that you want to run. You must also designate the location of the program by listing the slot and drive number where it will be found.

The RAM disks are set up in Slot 3. Programs are loaded onto drive number 1 (unless you choose to load them into number 2 or 3). So at the prompt you enter:

```
J RUN HELLO, S3,D1
```

(or whatever the name of the boot program is)

With DOS 3.3 disks, an easy way to identify the boot program is to use *Copy II Plus* and select the "Change Boot Program" option. This routine first identifies the current boot program. When the boot file name is displayed... write it down and use that name in the DOS 3.3 command line above.

Most DOS disks will probably be Hello but the name of the boot program can be any legal file name.

### Some programs will load; Some won't

You can use some programs but not others. A basic rule is that you can load any program that has standard DOS 3.3 or ProDOS format.

You cannot load any program that is copy-protected.

In Chart 2, some of the more popular programs are summarized.

Chart 2:

### A look At Several Programs

OP.SYS	Startup file	Can you load it
unknown	Locksmith 6.0 unknown	no
unknown	EDD unknown	no
ProDOS	Utilities for //c BASIC.SYSTEM	yes
ProDOS	Probyter (Beagle) BASIC.SYSTEM	no
DOS 3.3	CIA HELLO	yes
ProDOS	ProDOS users disk BASIC.SYSTEM	yes
DOS 3.3	DOS 3.3 system master HELLO	yes
ProDOS	Copy II Plus 6.0 UTIL.SYSTEM	yes

Admittedly this list is not complete but it probably contains most of the programs that the average computists will have in their Hackin' Shacks. The programs in this list that will not load are not standard formatted disks. They cannot be read by the Autocopy or the *RamDrive* programs.

Although *Probyter* is a ProDOS disk, the catalog is not standard and it will not load. If

you look closely at the catalogs for the Apple Utilities for the //c, you will notice that there is a STARTUP file on it. When starting this program, it is critical that you do not run this file. Running STARTUP will cause the disk to reboot if it is still in the drive. You must run the BASIC.SYSTEM file in RAM to use the program that you have loaded into memory.

EDD (Essential Data Duplicator) is copy-protected and cannot be loaded into memory. For EDD, this is not a problem because it loads completely into memory on bootup and gives the same fast access.

There is not much use in loading the Utilities for the //c into memory unless you put it in a subdirectory in RAM. Having it in a subdirectory however will let you access it very quickly if you are using it in conjunction with, for example, *Copy II Plus*. In some situations, it will be very handy to have several utility programs instantly available. Using the RAM drive will be of tremendous help at those times.

### One Users Evaluation

Is the speed and efficiency worth the cost? I wouldn't suggest that you buy a memory expansion board if all you do is work with simple utilities and make occasional copies or backups of files or programs. But being able to use utilities and copy programs with a memory expansion board is another reason you can add to that list you are keeping as justification to buy one. Add this reason to those like expanding *Appleworks* space, using CP/M, speeding up program execution with the new 16 bit processor and you probably have enough reasons to go out and buy one.

When you are thinking about how much time is saved, be sure to compare the time saved to any losses in speed. Remember how *Copy II Plus* knows that RAM is engaged? That does slow down the routine copy process.

There is a time to use it and a time not to use it. If all of your utility or copy program is routinely loaded into standard memory on boot up, this will be no help. Loading all of your program into memory is especially helpful when your program makes frequent trips to the disk for routines, overlays or data. Placing all of the program into the RAM Drive will eliminate all of the mechanical access to a floppy disk. In short, it is worth it to me.

In summary, the Applied Engineering group is one dynamic gang of people. With their memory expansion boards, they have given us a very useful utility that makes using the expanded memory with utility programs a super-easy process. By placing your utilities completely in memory, you can save significant amounts of time when working. The lightning fast RAM disk lets your computer do more and since it is faster, you will be faster. While this one use is probably not enough to justify the purchase of a board, it is one excellent reason out of many to go out and buy a memory expansion board. So go ahead and get one and, put your utilities where your RAM is!







when  
writing  
a...

## letter to the editor

- 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 also be placed in the RDEX section.

- Address your letters to:

COMPUTIST  
Editorial Staff  
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.

RDEX EDITOR

## input

The INPUT column has been made a part of the Readers Data Exchange forum.

Opinions expressed are not necessarily those of COMPUTIST, its volunteer editorial staff or SoftKey Publishing. The Readers Data Exchange is a forum for all readers where exchange of data, whether opinions, softkeys, copy parms, or general information, can occur.

John Lockard

I am a new subscriber to COMPUTIST magazine, and from the issues that I've gotten, I'm very happy with the subscription.

a note on Rob Hall's...

### Taking the Wiz out of Wizardry COMPUTIST 40, Page 23

Looking at the article *Taking the Wiz out of Wizardry* in issue 40, quite a bit of the article helped me out a lot. Being a fanatic with the *Wizardry* series, I was surprised by some of the things in the article.

### More Wizardry APTs

Here are some more *Wiz* APTs.

First, when changing characters, use the character back-up disk. For one, you can have as many backups of characters as you want, and secondly, the characters on the backup are formatted to the beginning of each sector, as follows:

CHAR.	TRK.	SEC.	CHAR.	TRK.	SEC.
01:	00	00	12:	01	09
02:	00	0D	13:	01	07
03:	00	0B	14:	01	05
04:	00	09	15:	01	03
05:	00	07	16:	01	01
06:	00	05	17:	02	00
07:	00	03	18:	02	0D
08:	00	01	19:	02	0B
09:	01	00	20:	02	09
10:	01	0D	1D:	02	07
11:	01	0B			

### No ID No.

The ID sector contains the volume of the *Wizardry* disk that you own. When you boot the *Wizardry* (PG, KOD, or LOL) you see the serial number, that's the ID number. The ID number is contained on each of the *Wizardry* disks on one of the first few sectors of either track \$00, or \$01. Changing that to 00000 on all *Wizardry* disks will make backing-up, and trading characters a lot easier.

### Golden Error

Gold is contained in bytes \$34 - \$39, not just \$34 - \$38.

### Experience Error

Experience is also 1 byte longer than reported, \$7C - \$81, not just \$7C - \$80.

### Aging

In LOL, a value for age: \$26/\$27 for 20 years old, \$01/\$04.

### Status

One byte \$28: status:

value	status
\$00	- O.K.
\$01	- afraid
\$02	- asleep
\$03	- paralyzed
\$04	- stoned
\$05	- dead
\$06	- ashes
\$07	- lost forever

### Curses!

Starting at \$3C, going all the way to \$7B, the third byte (ex. 3E, 46, 4E, 56, 5E, 66, 6E, 76) is the byte which tells if the item is cursed or not:

value	status
\$01	- cursed
\$00	- not cursed

### Spells

Mage and priest spells can also be given up to 255, just by changing every other byte starting at \$92 to \$FF.

### Character location

Location of your character is stored at EAST-\$C8 NORTH-\$CA, UP-\$CC, values of \$00/\$00/\$00 are at the castle.

### Awards and Honors

Awards (honors etc.) located at \$CE and \$CF are as follows.

\$CE	\$CF
\$01 "<" pg	\$01
\$02	\$02
\$04	\$04
\$08	\$08 "KD" KOD
\$10	\$10
\$20 "*" LOL	\$20 r/p LOL
\$40	\$40 "DG" KOD
\$80	\$80

So a value of \$21 at \$CE would give you both the **Chevron**, and the \* of LOL. While a \$20 at \$CF would give you the rite of passage in LOL.

Also, remember, you can make any combinations of characters, a good Ninja, neutral or evil Lord, etc.

Have Fun.

## 2 Greg Poulos

My friend, Willem, recently showed me how to crack *Guitar Wizard*. Here's the crack:

Softkey for...

### Guitar Wizard

**1** Copy the original with any normal, whole disk copier like COPYA.

**2** Run a sector editor and make these changes to track \$00, sector \$06:

Track	Sector	Byte(s)	From	To
\$00	\$06	\$2E	\$D0	\$EA
\$00	\$06	\$2F	\$D5	\$EA

**3** Write back the sector. You're done.

## 3 Brian Symonds

I just received issue COMPUTIST 41 and found the article on *Gemstone Warrior* (pg 14) very interesting. However, the softkey listed will **not** work on my version.

Softkey UPDATE for...

### Gemstone Warrior

COMPUTIST 41 page 14

I have version 1.1 which has the address field epilog bytes changed from DE AA to 9C C9. So, change step 1 in the softkey to ignore the address field epilog bytes as well as the data epilog bytes:

**1** Boot up DOS 3.3 and type the following:

**I CALL -151**

\* B934:18 60

\* B988:18 60

**2** remains the same.

**3** you must do a sector-edit to either change the address field epilog bytes to normal for reading and writing, or patch DOS to ignore them (which is what I chose to do!) So in step 3 add the following sector edit to the ones provided:

Track	Sector	Byte(s)	From	To
\$00	\$03	\$88	\$A8	\$18
\$00	\$03	\$89	\$D0	\$60

You could of course use *Super IOB* to copy the disk by capturing the *Gemstone Warrior* RWTS (if you have a means of breaking out of the program and into the monitor) and writing a NewSwap controller, and including the appropriate sector edits.

a Softkey note for...

### Notable Phantom

Designware

and for...

### Micro Wine Companion

*Notable Phantom* by Designware and *Micro Wine Companion* can be deprotected by capturing their RWTS (you have to be able to break out of the programs and into the monitor) and using a NewSwap controller with Super IOB.

A quick Softkey for...

### Stickybear Printer

Optimum

The *Stickybear Printer* by Optimum can be deprotected by using the method described in

COMPUTIST 15 (also, Book Of Softkeys Volume III, page 94) for *Stickybear* programs.

## 4 Michael David

I recently discovered a very useful educational ware from Grolier entitled *Note Card Maker*.

Basically this program permits children to be assisted in writing term papers through the use of note cards.

However, as with many Grolier software, this is a protected program, written in Pascal.

Softkey for...

### Note Card Maker

Grolier

#### Requirements:

- copier
- sector editor

Cracking it is not too difficult. You will need a quick copy program such as *Copy II Plus* or *Disk Muncher* and a sector-editing program such as *Copy II Plus*.

Although the disk will initially copy very quickly and appear error-free, it will not work. Here is how to crack it.

**1** Copy **both** sides of the program with the quick copying program of your choice.

**2** Use your sector editor and edit as follows:

Track	Sector	Byte(s)	From	To
\$1C	\$00	\$82	\$BD	\$60
\$1A	\$09	\$68	\$BD	\$60

**3** Write your sector edits to disk. That's all there is to it.

## Cracking other Grolier Wares

I have cracked other Grolier Wares using a similar method. Unfortunately, I did not record the exact changes that were made.

However, I have found that the quick copy method is quite nice if one has a NMI such as *Wildcard*. After the program is booted and "died", I simply press the *Wildcard* button,



enter the monitor and list the area where the program crashed.

A sector editor enables me to locate the track and sector where the problem bytes occur.

5

David Goforth

Well I've finally accumulated enough information to justify the \$.22 stamp to send this off.

First a couple of minor errors:

*Corrections for the Sofkey to...*

## Mousepaint

COMPUTIST 36, page 22

In COMPUTIST 36, page 22, the softkey for *Mousepaint* says to look for A9 00 00 on the disk, these should be A9 00 60 (at least it was on mine).

Also, scan the disk for 69 00 D0 03 and change the D0 03 to EA EA. This will disable the original Apple check by PRODOS. (I found these on track 1, sector 9, bytes 5B and 5C)

*a note on Wingfields'...*

## Electronic Disk Drive Swapper

COMPUTIST 39, Page 16

I installed the drive swapper shown in COMPUTIST 39 by William Wingfield and am glad to say it worked perfectly.

*a note on B. Snook's...*

## QuickPop into the Monitor

COMPUTIST 25, Page 5 Input

I also installed a 100 ohm resistor through a switch to reset into the monitor as described by Brian Snook in COMPUTIST 25 and I would like to give you a...

### \*\*\* WARNING \*\*\*

While it will reset into the monitor for most programs, if the DOS commands have been altered, they remain that way. So be careful.

If you decide to try to CATALOG the disk... something strange and unwanted may occur -

such as an INIT). I found this on two disks, the DOS commands CATALOG and LIST pointed to the INIT command (luckily my disks were write-protected).

Now for some softkeys...

*Softkey for...*

## Starcross

The only protection used is changed data prologues on most of the tracks (and sectors). It uses D5 AA BC (and D5 AA AD in some places). The controller below will copy the whole disk reading both data prologues and perform a sector edit on track 0, sector 2, byte FC, from BC to AD (this will allow the DOS to read all sectors with normal data prologues).

### Controller

```
1010 REM STARCROSS CONTROLLER
1010 TK = 0:LT = 35:CD = WR:MB = 151: ONERR
GOTO 550
1020 ST = 0:T1 = TK:GOSUB 490:GOSUB 270
1022 IF TK > 0 AND TK < 25 THEN POKE
47356,188
1024 IF TK = > 25 THEN GOSUB 230
1030 GOSUB 430:GOSUB 100:ST = ST+1:IF TK = 2
AND (ST = 2 OR ST = 4 OR ST = 6 OR ST = 15)
THEN POKE 47356,173:GOTO 1030
1032 IF TK > 0 OR TK = 0 AND ST = 16 THEN
POKE 47356,188
1034 IF TK => 25 THEN GOSUB 230
1036 IF ST > 16 THEN 1030
1040 IF BF THEN 1060
1045 IF TK = 24 THEN GOSUB 230
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 1070
1090 IF TK < LT THEN 1020
1100 HOME:AS$ = "ALL DONE":GOSUB 450:END
```

*Softkey for...*

## Wishbringer

Copy with *Super IOB* Standard Controller ignoring read errors (because of difficulty reading track 14, sector 2) and do not write-protect the copy.

*Softkey for...*

## Dinosaur Dig

Uses altered address and data epilogues, DF AB instead of DE AA. The controller below will read the altered marks and sector edit the following locations on the copied disk so DOS will be able to read the normal epilogues.

Track	Sector	Byte(s)	From	To
00	2	9E	DF	DE
00	2	A3	AB	AA
00	3	35	DF	DE
00	3	3F	AB	AA
00	3	91	DF	DE
00	3	9B	AB	AA
00	6	AE	DF	DE
00	6	B3	AB	AA

### Controller

```
1000 REM DINO DIG
1010 TK = 0:LT = 35:CD = WR:MB = 151
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 310: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 1070
1090 IF TK < LT THEN 1020
1100 HOME:AS$ = "ALL DONE":GOSUB 450:END
5000 DATA 213, 170, 150, 213, 170, 173, 223,
171, 223, 171
5010 DATA 8 CHANGES
5020 DATA 0, 2, 158, 222
5030 DATA 0, 2, 163, 170
5040 DATA 0, 3, 53, 222
5050 DATA 0, 3, 63, 170
5060 DATA 0, 3, 145, 222
5070 DATA 0, 3, 155, 170
5080 DATA 0, 6, 174, 222
5090 DATA 0, 6, 179, 170
```

*Softkey for...*

## Dam Busters

It uses altered address epilogues. Also, ignore any errors while reading.

Copy with Super IOB using the following controller:

Controller
1000 REM DAM BUSTERS CONTROLLER
1010 TK = 0:LT = 35:CD = WR:MB = 151
1020 ST = 0:T1 = TK:GOSUB 490
1025 POKE 47507,0:POKE 47517,0
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 TK=T1:ST=0:GOSUB 490
1065 POKE 47507,174:POKE 47517,164
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 1070
1090 IF TK < LT THEN 1020
1100 HOME:AS= "ALL DONE":GOSUB 450:END

## A Few Last Minute Notes

A few additions that may be incorporated into Super IOB (taken from past issues and very handy to have ready and available). *Note: This will change the checksum result of your program.* They may be incorporated anywhere you like, but I put them with the rest of the altered marker section as follows:

## Super IOB Additions

```
175 REM IGNORE ADDRESS EPILOGUES
178 POKE 47507,0:POKE 47517,0
215 REM IGNORE DATA EPILOGUES
218 POKE 47415,0 : POKE 47416,189 : POKE
47417,140 : POKE 47418,192 : POKE
47419,16 : POKE 47420,251 : POKE
47421,201 : POKE 47422,170 : POKE
47423,76 : POKE 47424,158 : POKE
47425,185
255 REM NORMALIZE IGNORES
258 POKE 47507,174 : POKE 47517,164 : POKE
47405,19 : POKE 47415,244 : POKE
47416,234 : POKE 47417,189 : POKE
47418,140 : POKE 47419,192 : POKE
47420,16 : POKE 47421,251 : POKE
47422,201 : POKE 47423,170 : POKE
47424,240 : POKE 47425,92
275 REM IGNORE DATA CHECKSUM
278 POKE 47405,0
```

## Some Helpful Charts

Also: enclosed is a helpful chart of Hexidecimal to Decimal (and back) conversions and how this is accomplished. This is very handy for the beginner or someone who gets tired of loading a program off disk everytime they want a conversion.

## Hexadecimal to Decimal Conversion Table

(VAL \*4096)+(VAL \*256)+(VAL \*16)+VAL= DEC

EXAMPLE:

BA65 = (11\*4096)+(10\*256)+(6\*16)+5=  
DEC 40960+2560+96+5= 43621

\$1= 1	\$10= 16	\$100= 256	\$1000= 4096
\$2= 2	\$20= 32	\$200= 512	\$2000= 8192
\$3= 3	\$30= 48	\$300= 768	\$3000=12288
\$4= 4	\$40= 64	\$400=1024	\$4000=16384
\$5= 5	\$50= 80	\$500=1280	\$5000=20480
\$6= 6	\$60= 96	\$600=1536	\$6000=24576
\$7= 7	\$70=112	\$700=1792	\$7000=28672
\$8= 8	\$80=128	\$800=2048	\$8000=32768
\$9= 9	\$90=144	\$900=2304	\$9000=36864
\$A=10	\$A0=160	\$A00=2560	\$A000=40960
\$B=11	\$B0=176	\$B00=2816	\$B000=45056
\$C=12	\$C0=192	\$C00=3072	\$C000=49152
\$D=13	\$D0=208	\$D00=3328	\$D000=53248
\$E=14	\$E0=224	\$E00=3584	\$E000=57344
\$F=15	\$F0=240	\$F00=3840	\$F000=61440

## Decimal to Hexidecimal Conversion Table

Use the Hex-to-Dec Table to find the highest value that will divide into the number, then subtract that value from the original and continue to the next step.

DEC/4096 = WHOLE VALUE  
REMAINDER/256 = WHOLE VALUE  
REMAINDER/16 = WHOLE VALUE  
REMAINDER

EXAMPLE

62998/4096 = F (62998 - 61440)  
1558/256 = 6 (1558 - 1536)  
22/16 = 1 (22 - 16)  
= 6  
62998 = \$F616

## Scott Adams Graphic Adventures S.A.G.A.

So I bought two graphic adventure games: SAGA #2- Pirate Adventure and SAGA #3- Mission Impossible. (SAGA stands for Scott Adams Graphic Adventure). I am a fanatic about graphic adventures.

Speeding home in my car, I was anxious to begin tackling these adventures. I booted up SAGA #2 and began the adventure.

Everything was fine until I tried to 'get' an item. I could move from room to room, but as soon as I tried to pick up an item, the disk drive would start and stop three times then reboot.

I booted up SAGA #3 but had the same disappointing results.

## ON SALE! No Returns. No Refunds

Well, I immediately took the software packages back to the store demanding a refund. But the sales person in the store told me that "ON SALE" items were not returnable. Arguing with the manager didn't help either.

Well, with no other choice, I thought I would call Adventure International and ask for a replacement disk thinking that the disks were probably bad. After many telephone calls to the long distance operator, I found out that Adventure International is in Chapter 7 bankruptcy proceedings. (So that's why the software was on sale).

Softkey for...

## Pirate Adventure

Adventure International

You can imagine my frustration by now. Giving up, I thought I would catch up on some reading. I was reading through old COMPUTIST magazines and there it was!

In COMPUTIST 33, page 7, Edward Hauff had a softkey for SAGA #1 which described the exact symptoms that I had experienced. He mentions changing \$FF to \$00 at \$6B16 of file M2 in memory.

This change defeats the original disk check. What I can't understand is that my disks are the originals

## SAGA 2 & 3

For SAGA #2 and SAGA #3, the location to be changed is \$6AEF which is different than SAGA #1.

**1** Boot up the disk with a fast DOS and INIT both sides of the disk with the name HELO (spelled wrong is right!) and delete it.

## 6 Franco Panizzon

Yes! It's finally happened. I've cracked my first disk.

Well, it all started when I was visiting the local computer store and found that they had a software sale.



**1 INIT HELO**  
**1 DELETE HELO**

**2** Enter the monitor and disable DOS error checking.

**1 CALL -151**  
**\* B942:18**

**3** BRUN FID and copy all files on both sides of the original to both sides of the INITIALIZED disk.

Use the wildcard "=", when asked for FILENAME and answer with "N" when asked for PROMPTING.

**\* BRUN FID**

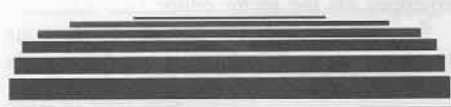
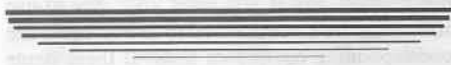
**4** Load M2 from the boot side. Enter the monitor and change byte \$6BEF from \$FF to \$00.

**1 BLOAD M2**  
**1 CALL -151**  
**\* 6BEF:00**

**5** Save the file back to disk.

**\* BSAVE M2, A\$6860,L\$0DF0**

Thank you COMPUTIST and Edward Hauff for saving my small investment.



**7** Bryan Sunda

*Softkey for...*



- Requirements:**
- Apple II 64K or better
  - COPYA
  - Sector Editor
  - 2 blank disks

The first thing I did was try and copy the disk with *Locksmith Fast Copy*. Track \$00, sectors \$00-\$09 were able to be copied. The rest of the disk was in nonstandard format.

The next thing I did was disable DOS's read error checking routine (B942:18) and used COPYA to copy the disk. I now had the disk in a standard format.

Now to figure out how the original was different from standard...

I used a nibble editor to check the prologue and epilogue of the address and data fields and found they were standard.

## The Protection: Nibble Count

I then decided to go to my sector editor and patch the DOS figuring out what needed to be changed in order to read this disk.

I found that just disabling the checksum would allow the disk to be read. So I disabled the checksum routine and booted the disk.

I then found they were using a nibble count. I did a little boot tracing and found where it was and disabled it.

The disk worked perfectly after that.

## The De-protection: Step-by-step

**1** Run your COPYA program and break out of it with [C].

**2** Now delete line 70 and disable DOS's read error checking routine (B942:18) by typing:

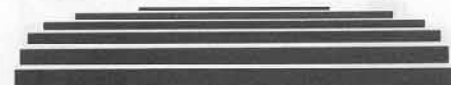
**1 70**  
**1 POKE 47426,24**  
**1 RUN**

**3** Copy both sides of *Infiltrator*.

**4** Finally make the following sector edits to side one:

Track	Sector	Byte(s)	From	To
\$00	\$03	42	38	18
\$00	\$08	EA	20	EA
\$00	\$08	EB	00	EA
\$00	\$08	EC	02	EA

Side two needs nothing more than being copied with DOS's error checking routine disabled (B942:18). Have fun...



**8** Dudley Brett

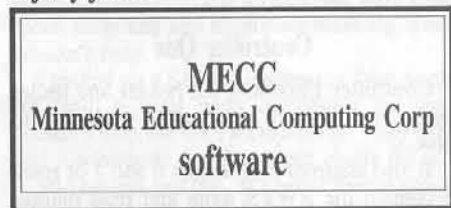
The Minnesota Educational Computing Corporation (MECC) has provided educational programs of merit for several years to schools.

Wisely, MECC allows school boards the license to provide unlimited copies to individual teachers. Thus a teacher may legally use multiple copies in a classroom with several computers.

Though this enlightened practice provides lower cost software and ensures instant backups can be provided at the time of the inevitable disk crash by gummy little fingers, MECC does use a simple, but effective protection upon its software.

Each copy usually has to be made by a bit copier, a device not necessarily available to the average teacher. Besides, a bit copy can be accomplished s-o-o slowly, and multiple class copies take forever.

*Softkey for...*



The answer to this problem is, of course, to remove the copy protection in order to make the disks CopyAble. This is relatively easy to do if one knows which protection scheme is being used.

## Early (1981) Protection

MECC disks initially had no protection, but in 1981 started employing simple changes of address and data header bytes. The standard approach of deprotection for this is to capture the RWTS and employ either a swap controller and Super IOB to copy tracks 3 to 22 to an initialized disk, or use DEMUFFIN PLUS to transfer files to an initialized disk. As MECC disks of this period often employed a slow DOS, they could be improved by the use of a fast DOS.

## Current (1984) Protection

In 1984, however, MECC started to employ routines within the programs to check for valid bytes in the DOS in memory. This is most commonly done by a machine language module loaded in at the end of the HELLO program. (When HELLO is run, the last line generally CALLS into this module). This module then loads several files, makes a check of DOS and then exits to the basic MENU.

## Deprotection Time

As I suspect that simply finding and turning off the DOS check in this module will not necessarily remove all possible checks, I believe

a method to retain most of the MECC DOS on a deprotected disk is the most rational approach.

On MECC DOS 3.3 disks of the period 1984 to present, most disks have on tracks \$1 to \$22 the address header altered from D5 AA 96 to AA D5 AD and the data header altered from D5 AA AD to D5 96 AA. Track \$0 has normal headers.

They also employ the previously mentioned DOS check.

To defeat this protection, track \$0 is copied normally and tracks \$1 to \$22 must be read with the "strange" bytes and written with "normal" bytes.

*Super IOB* v1.5 is, of course, a natural for this task.

I am enclosing two controllers which basically do the same job.

## Controller One

Controller 1 assumes the header and trailer bytes used will not be changed from disk to disk.

It also assumes that sectors 6 and 7 of track 0 contain the RWTS write and read marker bytes.

The sector edits are as follows:

Track	Sector	Byte(s)	From	To
0	6	86	96	AA
0	6	8B	AA	AD
0	7	1F	96	AA
0	7	2A	AA	AD
0	7	83	AA	D5
0	7	8D	D5	AA
0	7	98	AD	96

As I have not seen every MECC disk of the recent past, I cannot suggest every MECC disk will be deprotected this easily.

## Controller Two

I am including controller 2 which, using the RWTS trapped from a MECC disk, will copy track \$0 normally, will change abnormal read bytes to normal on track \$1 to \$22, but will make no sector edits.

The ingenious reader can, by using a sector editor on track \$0, find and replace goofy bytes with normal ones. Hopefully, these may even be on sector 6 and 7! Otherwise, use the search facility of the sector editor to find the RWTS. Try the sequence BD 8C C0 10 FB and then look around.

## Report MECC ProDOS Protection To COMPUTIST RDEX editor

Look for new protection schemes from MECC as they start to use ProDOS. I have seen

an educational database series which has this new method, but have as yet not looked into it. I would welcome any information from readers.

## Controller 1

```

1000 REM MECC CONTROLLER
1010 TK = 1: LT = 35: ST = 15: LS = 15: CD = WR:
      FAST = 1: POKE 48573, 128
1020 RESTORE : GOSUB 190: GOSUB 210: GOSUB
      170: GOSUB 490: GOSUB 610
1030 GOSUB 230: GOSUB 490: GOSUB 610: IF PEEK
      (TRK) = LT THEN 1050
1040 TK = PEEK (TRK): ST = PEEK (SCT): GOTO
      1020
1050 TK = 0: LT = 1: ST = 15: LS = 15: CD = WR:
      FAST = 1
1060 GOSUB 490: GOSUB 610: GOSUB 310
1070 GOSUB 490: GOSUB 610: IF PEEK (TRK) = LT
      THEN 1090
1080 TK = PEEK (TRK): ST = PEEK (SCR): GOTO
      1060
1090 HOME: PRINT "COPYADONE": END
5000 DATA 170, 213, 173, 213, 150, 170, 222,
      170, 222, 170: REM STRANGE HEADER AND
      TRAILER BYTES
5010 DATA 7 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
    
```

## Controller 2

```

1000 REM MECC SWAP CONTROLLER
1010 TK = 0: LT = 1: 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 TK = 1: LT = 35: ST = 15: LS = 15: CD = WR:
      FAST = 1
1060 GOSUB 360: GOSUB 490: GOSUB 610
1070 GOSUB 360: GOSUB 490: GOSUB 610: IF PEEK
      (TRK) = LT THEN 1090
1080 TK = PEEK (TRK): ST = PEEK (SCT): GOTO
      1060
1090 HOME: PRINT "COPYADONE": END
10000 REM CONFIGURATION TIME
10010 PRINT CHR$ (4); "BLOAD MECC.RWTS,
      A$1900"
    
```

9

Bryan Ferrel

## Taking More Of The Wiz Out Of Wizardry

I would like to make a few comments in regard to the article 'Taking the Wiz out of Wizardry' in *COMPUTIST 40* by Bob Hall.

First, I would like to commend him on a very complete and thorough job well done. I know that the work was very time-consuming because a friend of mine and I spent two days and nights figuring the same thing out two summers ago.

### Wizardry equipment

What I would like to add is in regard to the equipment. In the eight byte string that represents equipment, bytes three and four will show up \$FF FF if an item in that place is equipped and cursed. I don't know if this is significant or not but who can tell.

Also he says to be sure the equipment is unequipped. But if you want your guy to use something he's not allowed to use or equip the corresponding equipment slots. If these items are ever unequipped they will have to be re-equipped via the sector editor.

Last but not least I would like to say your magazine is the best there is and I hope it's around for years to come.



I certainly hope so, too. RDEXed.

10

B. Dudley Brett

Enclosed are two softkeys that might be of interest to readers. These are the result of my peculiar hobby...

## Seek Out and Eradicate New Protection Schemes

I enjoy the intellectual stimulation of finding out and eradicating the protection schemes employed on current software, not really because I have to use the disks, but just for fun!



I use very few commercial disks in my personal work. I usually design my own programs, believing that they fulfill my planned functions in a more direct way than any commercial program could do.

Besides the included more explanatory softkeys, your readers may be interested in some more quick softkeys. These are for disks presented to me by friends knowing my peculiarity.

Softkey for...

## Banner Catch

Scholastic

This has the same protection as *Story Tree* (COMPUTIST 37, page 8).

**1** From the Monitor:

\* B942:18 60

**2** Make a copy with COPYA.

**3** Make the following sector edits to the copy:

Track	Sector	Byte(s)	From	To
\$00	\$05	\$93	C6	4C
		\$94	2A	86
		\$95	D0	02

**4** Don't forget to write the changes back to your disk!

Softkey for...

## Turtle Tracks

Scholastic

This is essentially unprotected. From the monitor:

\* B942:18 60  
\* RUN COPYA

Softkey for...

## PFS File

Scholastic School Version

**1** Make a copy using COPYA.

**2** Make the following sector edits.

Track	Sector	Byte(s)	From	To
\$1E	\$0A	\$49	04	29
\$15	\$09	\$4D	0E	33

**3** Don't forget to write the changes back to your disk!

Softkey for...

## Microzine #12

## Microzine #13

## Microzine #14

This is not really original, but it may be useful...

Do the same as *Banner Catch* (Compliments of Phil Pattengale COMPUTIST 27, page 13) or try the following:

**1** INIT enough disks with Pronto DOS (more room than DOS 3.3!)

**2** Trap the RWTS from any *Microzine* Boot disk.

**3** \* BSAVE RWTS.Microzine, A\$6800, L\$800

**4** \* BLOAD DEMUFFIN PLUS, A\$4000

**5** \* BLOAD RWTS.Microzine, A\$6800

**6** Type the following:

\* 803<4000.6000M  
\* B800<6800.6FFFFM  
\* 803G

**7** Follow the prompts and copy all files!

Note that the second side of *Microzine #14* gets a DISK FULL error. Just copy it with COPYA.

George Cawthorne

## Does Anybody have schematics for The Apple Duo-drive?

Would you or any of your readers know where I could find the actual schematic diagram of the Apple Duo-drive?

I called Apple Computer in California and they acted like I wanted some top secret information! I talked to some Apple service centers around here and all they had were some block diagrams and alignment manuals. They couldn't help.

I picked up a SAMS schematic of the Apple //e from a local parts distributor but they couldn't help me with a Duo-drive. They only had a schematic for the Apple single drive.

So I bought that one. It's better than nothing. However, I still want the correct manual.

I even called the "Howard Sams" headquarters and asked them if they were planning on coming out with one. They had no plans at that time to do so.

Greg Poulus

*Marble Madness* is an excellent game at the arcades which is easily addictive. After waiting for several months for an Apple version, it finally came. *Marble Madness* is from Electronic Arts, and they did an excellent job on the game. And here is how to crack it.

Softkey for...

## Marble Madness

Electronic Arts

First of all, the disk is double-sided, although side B doesn't really seem to have any significant code, but I haven't played the game far enough yet to get to side B.

Both sides are copyable with any fast copy program, although you'll get an error at track \$06, on side A. This is the track that the protection scheme checks for the nibble count.

To this date, all of Electronic Arts' protection schemes are initiated with a 4C 69 xx (where xx is the page where it is). In *Marble Madness*, as in many other EA wares, it's at \$A000. So to initiate the protection scheme, the program does a 4C 69 A0 (JMP \$A069).

I scanned side A for these bytes (4C 69 A0) and found them in two places on track \$01, sector \$0F. At first all I did to blow these away was put an 18 60 in their place. But the guys at Electronic Arts are smart, they have the protection code check to see if these bytes have been tampered with by comparing the total value of all three bytes added together. So if you add 4C and 69 (in the monitor) and then that result plus A0, you have what is needed as a final number to balance. So all I did was put an 18 60 DD in the place of the 4C 69 A0's. I booted the disk, it bypassed the nibble counts, but then hung. Then after snooping through a lot of code, I found a JMP to \$569 which performs another check to see if the disk is the original. All that needed to be done here was to place a 60 on the first byte, which is a 4C, and the disk booted fine.

## Step-by-step

**1** Copy both sides of *Marble Madness* with any fast copier that ignores errors. Ignore errors on track \$06, side A.

**2** Run your favorite sector editor and edit the following bytes:

Track	Sector	Byte(s)	From	To
\$01	\$0F	\$00	\$4C	\$18
\$01	\$0F	\$01	\$69	\$60
\$01	\$0F	\$02	\$A0	\$DD
\$01	\$0F	\$6F	\$4C	\$18
\$01	\$0F	\$70	\$69	\$60
\$01	\$0F	\$71	\$A0	\$DD
\$01	\$0C	\$00	\$4C	\$60

**3** That's all. Tracks \$3, \$4, \$6, \$21, and \$22 are empty in case you want to put a title screen in the program or something like that.

### 13 Mario Chrsanthou

First, I would like to congratulate you on a very good service you are offering your subscribers.

## //gs & Copy II Plus

For three years I have been a proud owner of an Apple //e, and COMPUTIST has been a part of it ever since I decided to subscribe to your magazine.


Like many Apple hackers we have been waiting for the release of the Apple //gs. Upon it's arrival, I immediatly sold my Apple //e to my brother and purchased a new Apple //gs.

Now as an owner of an Apple //gs, COMPUTIST has become invaluable to my brother, but of no use to me as all software for the Apple //gs comes on a 3 1/2" inch disk. My copy utilities are not yet able to copy or use most of the copy utilities available.

I have contacted Central Point Software and asked them if they had any plans to upgrade Copy II Plus for the Apple //gs and the Apple 3 1/2" drive. Happily their reply was that because of the difference of the Apple 5 1/4" drive and the Apple 3 1/2" drive their programmers will have to do a complete rewrite.

## Will COMPUTIST cover //gs?

Now the most important question is will COMPUTIST follow Central Point Software and keep on top while keeping the older Apple II's compatible? If so then you will still be a part of my utilities. If not then good luck!

 Because of the open forum format of the Readers Data Exchange column, we can easily publish the most up-to-date cracking methods for ANY computer system. Readers need only send in the info and we'll print it. RDEXed

### 14 Eric W. Taylor

When I first saw *Writer Rabbit*, I thought it was a very good educational program. Then I saw the softkey for *Math Rabbit* in COMPUTIST. I thought they might be the same, but the *Math Rabbit* softkey did not work on *Writer Rabbit*. So I started snooping around.

Softkey for...

**Writer Rabbit**  
Learning Company

I was able to fast-copy both sides of the disk with *Locksmith Fast Copy*. I then tried to boot the disk. The introduction page came up, and I thought I was done. But it turned out that I was not. I pressed a key to continue and the drive came on. The dreaded message "Fatal Disk Error" came up. I then knew there must be some sort of nibble count or checksum so the program could verify the presence of an original disk.

## Nibble count or checksum?

I then decided to search the disk for any sort of disk activity. I set up my disk searcher to look for the sequence \$C0 \$8C. This sequence is the set used in the instruction LD \$C08C,X. This is the sequence to load a byte off of the disk.

I searched the entire disk. There were some disk accesses in the DOS area. The interesting accesses occurred on the track three and track ten on the first side. On the second side this sequence occurred on track eight.

I looked at these routines and found the beginning of them. This was easily done by lookingh back from the disk access sequence until I found an RTS. The RTS is at byte \$60. I assumed the bytes after the RTS were the first instructions in the disk check routine. These bytes were at \$BD and at \$89 in all of the routines.

I then revised my search pattern to \$60 \$BD \$89. I searched both sides for this pattern, and when I found the occurrences I replaced the \$BD \$89 with a \$18 \$60.

I wrote these sectors back to the disk and then I wrote the disk. This time the disk booted all the way through. I checked the back side, and it was deprotected as well.

## The Cookbook Method

**1** Copy both sides of *Writer Rabbit* using a copier which will ignore errors. I used *Locksmith 6.0 Fast Copy*.

Now put away the original.

**2** Using a disk-search utility, search the copy for the sequence \$60 \$BD \$89. Replace the \$BD \$89 with \$18 \$60 and write the sector back to the copy.

## Search both sides thouroughly

Make sure you search the entire disk and both sides. On my copy the occurrences of \$60 \$BD \$89 were:

Side	Track	Sector	Byte
A	\$03	\$00	\$8A
A	\$10	\$09	\$DB
B	\$08	\$03	\$DB



So the corresponding changes I made were:

Side	Track	Sector	Byte(s)	From	To
A	\$03	\$00	\$8B	\$BD	\$18
A	\$03	\$00	\$8C	\$89	\$60
A	\$10	\$09	\$DC	\$BD	\$18
A	\$10	\$09	\$DD	\$89	\$60
B	\$08	\$03	\$DC	\$BD	\$18
B	\$08	\$03	\$DD	\$89	\$60

That is all there is to do the softkey *Writer Rabbit* from the Learning Company.

I hope the process has taught the beginners a little about the steps to use to go about softkeying a disk. Have fun with the procedure, maybe some other Learning Company programs are protected in the same manner.

I hope this softkey is useful for your magazine. I really appreciate your work with *COMPUTIST*, and I hope it always remains a quality magazine.

## 15 Donnie Cambre & Johnny Garret

*Arcticfox*, the futuristic tank simulator from Electronic Arts, is very similar to the arcade hit *Battle Zone*. Your mission, should you choose to accept it, is to destroy the alien race that is attempting to take over the Earth. This alien race has built a fortress in Antarctica and is currently removing the oxygen from the atmosphere.

Softkey for...

### Arcticfox

Electronic Arts

#### ■ Requirements:

- 64K Apple II
- COPYA* or *Locksmith Fastcopy*
- A sector editor

To remove the protection from *Arcticfox* you must have a basic understanding of how Electronic Arts protects all of its software.

The protection code consists of a strangely formatted track 6, as is common with most other EA releases. EA does a series of check routines that can be heard by the distinctive clatter that the drive head makes when reading track 6.

These check routines cannot be defeated by NOP'ing the routine call because the program checks for the generated checksums.

If these checksums are not correct then someone has tampered with the code and the program will bomb. You can usually find these checksums by searching for the following bytes:

4C 69 A0 4C 69 05 20 F8 XX 20 03 A0  
(XX can be nearly any value.)

These sequences usually indicate a call to the protection routine, however occasionally these are valid calls to the program itself. The easiest way to determine this is to get a disassembly of the sector and look for suspicious JMPs, JSRs, LDAs, STAs, and most especially CMPs.

Generally the protection code is found on track 1, sectors A-F.

### Step by step

**1** Copy disk with *Locksmith Fastcopy* or use *COPYA* with the error checking defeated as follows:

**J** RUN COPYA  
**J**  (at prompt)  
**J** CALL -151  
 \* 3A1:18  
 \* 3D0G  
**J** RUN 80

**2** Take out your favorite sector editor and change the following bytes:

Track	Sector	Byte(s)	From	To
01	0B	47-49	20 F8 05	18 60 40
01	0E	47-49	20 F8 A0	18 60 40
01	0E	4C-4E	20 03 A0	18 60 4B
01	0F	00-02	4C 69 A0	18 60 DD
01	0F	6F-71	4C 69 A0	18 60 DD

**3** Place your original in a safe place. Go ahead, lose your restraint!

### Final Notes

In the future, we would appreciate it if the authors of softkeys would show what the "FROM" bytes were originally because not all of the games have the same protection. This would prevent confusion.

## Turn *Karateka* Into A 'WIRED' Game

For those of you who like "WEIRD" games, like I do...

Take your original *Karateka* disk and boot up the backside of the disk. It may make the game a little more difficult but it makes it very interesting.

## Airheart protection

I have also purchased *Airheart*, Broderbund's newest release, and the graphics are astounding and as usual the protection isn't bad either.

I can succeed in making a workable copy but the game will freeze after doing a "disk check" at the end of the game and the only way to play again is to reboot the disk.



16

Charles Taylor

*Age of Adventure* is a reprint of two arcade graphics adventure games that were good when they came out, but are now well below Electronic Arts' usual high standards.

The two games are *Ali Baba and the Forty Thieves*, and *The Return of Heracles*, each side containing one game.

Softkey for...

### Age Of Adventure

Electronic Arts

#### ■ Requirements:

- A two-sided (notched) disk
- A sector editor/disk searcher
- A copy program that can ignore errors

The copy protection is the standard Electronic Arts protection, consisting of a single hard-to-copy track (track six).

The deprotection is also standard, consisting of searching the disk for 4C 69 A0 and changing it to 18 60 DD.

**1** Copy both sides of the disk with a whole disk copier that can ignore errors (or skip tracks).

**2** Using a sector editor, scan the disk for 4C 69 A0, and change that sequence to 18 60 DD.

I found this sequence on both sides at track 1 sector F twice, once starting at byte 00, the second starting at byte 6F.

17

M. Ferreira

Let's get started with the good stuff, then we can go straight to the gripes, moans, comments and whatever department!

Another Sofkey for...

## Infiltrator

Mindscape

Run COPYA.

Hit **⌘C**

**I** CALL -151

\* B942:18

\* BE48:18

\* 3D0G

**I** 70

**I** RUN

When the copy is finished get out your sector editor and search for these two sets of instructions: 20 0D BE and 20 00 02. Change them to EA EA EA.

Read in track 0, sector 3. Change byte 42 from 38 to 18.

Be sure to copy the back side in the same manner.

### Back Side Is NOT Verified

Having been unable to progress in this game, I cannot verify the back side, but was told by the publisher it contains the third mission!!! Good Luck.

Now for that other category:

*Infiltrator* is a combination of helicopter simulation and adventure. The problem I encountered was the inability to progress beyond flying to the adventure portion! After nearly 75 times in the air, all ending in disaster of one kind or another, out of gas, being shot down, improper landing, (which by the way could have its' own sub-directory), or who knows what!!!! Maybe I just wasn't cut out to be a pilot, but for my money this game is no game. It is frustrating and almost totally unplayable!!

### Please, No More Antique Sofkeys

In regards to softkeys, personally I would like to see more accent on the newer releases as opposed to the Antiques. An occasional one wouldn't be objectionable, but it might be more beneficial to run a machine language tutorial than to fill up valuable space with softkeys for programs that have been cracked for ages.

**I** Because we have eliminated the time-consuming editing/verifying process that made some softkeys obsolete before their printing, and because we have the ability to print every softkey submitted to RDEX almost immediately, it will no longer matter if some are for historical or forgotten software. RDEXed

18

Bob Hazelwood

*Might & Magic* is a fantasy role-playing game that should be a map-maker's delight! The game has five cities, six castles, numerous caves and caverns, and twenty outside sectors, all of which have to be mapped by the player! Invest in plenty of graph paper!!

The program comes on both sides of two diskettes, but only the first side of the first diskette is protected. Being a relative novice to de-protection, I thought I'd take the easy way out and use *Copy II Plus* to make a duplicate of the boot-up side. WRONG!

The disk would boot ok, but as soon as the title page appeared, the drive re-booted! I had read in *COMPUTIST* that this often indicated a nibble count was being used, so I re-copied it using the KEEP TRACK LENGTH option. Unfortunately, I got a Nibble Count error on track \$0D. You guessed it, that's the track the program checks!

Sofkey for...

## Might And Magic

Well, if they're going to force me to try and de-protect the disk, I'll sure give it a shot!

I used a disk scan and found the JMP \$C600 (re-boot routine) on track \$00 sector \$0F.

Using the sector editor of Copy ][+, I replaced it with a JMP \$FF59 into the monitor. (That's another trick I read about in *COMPUTIST*).

So I re-booted my copy, got the expected \* prompt, looked in memory to find the routine and ... nothing.

The memory was full of \$00's!

The routine before the re-boot JMP puts \$00's in memory. Ok, I thought, I'll just NOP this code so that the program would remain intact.

### Now we're cooking!

Re-boot, JMP to the monitor, and find the code. The routine was located at \$036E.

Simple, I said to myself, I'll just search for references to this location. Since I don't (yet) have a way to scan memory (I don't think I do, anyway. I am just beginning, after all).

I used the disk scan to look for the code.

No such luck... On a lark, I replaced the JMP \$FF59 (formerly JMP \$C600) with a NOP NOP RTS (EA EA 60). Guess what? It works!

To summarize:

### The Cookbook

**1** Make a copy of side A using a copier that ignores errors.

**2** Use a sector editor and make the following changes:

Track	Sector	Byte(s)	To
\$00	\$0F	\$7A-\$92	EA (NOP)
\$00	\$00F	\$93	60 (RTS)

**3** Plan on doing a lot of mapping!

19

Jim S. Hart

The *Print Shop* and *Newsroom* articles on how to add printer drivers were most helpful. I can now 'custom design' those two programs for most any printer-interface combination.

### What's wrong, IBM?

In regards to the gentleman who thinks that Apples are not going to be part of the future of microcomputers: all I can say is HA HA. IBM has consistently been behind the times and they are always quick to imitate, whereas Apple has been and will continue to be the innovators in the industry.

IBM has just recently started to use 3.5" disk drives while the Mac (and Lisa) had been using them for years.

The window style interface also has been standard for years in the Apple camp while it is just coming into play for the IBM. What's wrong IBM? It sure takes awhile for you to rediscover and reinvent the wheel.



IBM supporters will be quick to ask why Apple is now trying to give the II and Mac lines MS-DOS compatibility. Simple. IBM has been a leader in mainframe systems for many years and businesses quite naturally assumed that this excellence would permeate down to the IBM microcomputers. This is why the IBM micros have such an abundance of business software.

Talk to any technical person about the quality of the hardware, however, and you discover that of all the MS-DOS machines, IBMs are not even close to being the leaders.

Heck, they can't even write their own operating systems for their machines. Microsoft had to come to the rescue with their fine MS-DOS.

When the new 886 IBM come out, there won't even be an operating system for it (guess why?).

Apple wants a piece of the business market and to get it, they have to go along with the standard.

With the introduction of the new MacIntosh II, Apple will be putting out the most powerful personal computer on the market for years to come.

Yes, I think that Apple has a future in this business of ours, if for no other reason than they are the innovators, not the imitators.

Now for a couple of short notes.

## Softkey for...

### Space Station

*Space Station* is a slick little program recently put out in which you actually get to control a space shuttle and build a space station, among other things.

To deprotect it, follow these short steps:

1

1 RUN COPYA

2

1 (C)

3

1 POKE 47426, 24

4

1 70

5

1 RUN

On the copied disk, use something like *Copy II Plus* to put normal DOS on the disk.

## How To Make MicroSparc's Assembler and ProDOS Macrosoft Run On A Laser 128

I recently had the chance to try out MicroSparc's *Assembler & Macrosoft* (ProDOS version). I have a Laser 128 (ROM version 2.9) and I quickly found out that the *Assembler* would not run on my computer. It does run on my friend's //e, however, so it wasn't an I/O error on the original.

This prompted me to find out what the deal was. Booting my *ProDOS User's Disk* and then getting into BASIC, I took a look at the *Assembler's* directory.

The first .SYS file on the disk's directory (other than ProDOS itself) was BOOT.SYSTEM, so that's where I looked around:

1 CALL -151

\* BLOAD BOOT.SYSTEMS, A\$2000,  
TSYS 2000L

The second instruction encountered is a JSR \$210E so that's where to go next:

\* 210EL

Examining this bit of code reveals it is responsible for printing out the title screen and copyright notices.

At \$2118 you'll notice a JSR \$FDF6, up into the ROM area. This caught my eye quickly because since the Laser's ROMs are different from the Apple's (out of necessity), this JSR may go to a subroutine that 'is not there'.

Pulling out my APPLE REFERENCE MANUAL, I looked through the autostart ROM listing in the back of the manual and BINGO, I found the bug.

The routine at \$FDF6 (from the program's point of view) is supposed to print out the character in the accumulator. The standard routine to do this is at \$FDED (COUT).

It seems that the author made a jump to a non-standard entry point (\$FDF6) instead of the standard entry point (\$FDED) to print out the character in the accumulator. I changed the code so that it now went to the standard entry point and saved the program back to disk:

\* 2118:20 ED FE (WAS 20 F6 FD)  
\* BSAVE BOOT.SYSTEM, A\$2000,  
TSYS

This change now allowed the *Assembler* to work just fine on my Laser and I then had the chance to check it and *Macrosoft* out. Both are solid values for the programmer. I have since given the originals back to my friend and am trying to decide between *Assembler / Macrosoft* and *ZBasic*, which will get my software dollars. I'll probably end up getting both.

20

Ronald E. Keeton

## I Need Help In Solving...

### Legacy of Llylgamyn

I desperately seek your help in aiding me in my efforts to solve the *Legacy of Llylgamyn*. I would like a clue or answer to the problem that appears: **Not Without Payment!**

Please help!! Surely someone knows the answer to this problem and will be kind enough to jot down the solution.

## Adventure Tip for...

### Zork II

As my offering to your tip section, I give the clue for *Zork II*...

● *The basket and canvas can be inflated to take you to new heights!!!*

21

Frank Villaroman

## Calling Miss Ann Onymous

I wonder if it's possible for you to put me in touch with Ms. Ann Onymous, your reader who wrote the letter about *Keyperfect*.

I would like to obtain a copy of the utility she used called "ProDOS Diagnostic Wedge". I've a lot of software I could swap in return. *a note on Nibble's...*

### Key Perfect

It's clear from her procedure that *Nibble's Keyperfect* is now a ProDOS disk. The one I have version 4, by the way--was on a 13 sector disk. It had normal marks, except for the data epilog which was BB AA. I got it filed in a normal 16 sector disk, monitor and changed byte 1A63 from 38 to 18. Then entered 803G to get Muffin started; and that was it. The file was only 33 sectors long. Thanks in advance.

22

Leigh Rowan-Kelley

I tried the Softkey for *Alternate Reality* in COMPUTIST 38, and found that it did not work at all for my copy.

A BUG in Lau's Softkey for...

## Alternate Reality

COMPUTIST 38, page 19

I had already found some of the altered checksum marks that Stephen Lau had, and I also had found the altered RWTS at \$800 when I started Boot Tracing it.

His controller seemed reasonable so I used that as a starting point. There were some errors in that however.

Line 5010 should be:

```
..... EA'' :GOTO 5040
```

rather than:

```
..... E: GOTO 5040
```

And line 10010 should read:

```
10010 PRINT CHR$(4)'' BLOAD ALT.RWTS,  
AS1900"
```

This was now going to work but first we had to get a copy of the RWTS. To get that, simply boot the disk, wait till it finishes loading in and you have the first text screen, then reset to monitor.

Next move the RWTS routine from its present location to a safe place the RWTS as Stephen suggested. You will find this works!

In Cookbook form:

**1** Run Program

**2** Reset into the Monitor via Reply/Integer/ NMI etc.

**3**

\* 1900<800.FFM FFFM

**4** Boot the Slave Disk.

\* 6[=P]

**5** Re-enter the Monitor

**1** CALL -151

\* 60

\* 2900<B800.BFFFM

**7**

\* 2B29<1B29.1BFFM

**8**

\* [C]B

**9**

\* BSAVE ALT.RWTS,A\$2900,L\$800

**10**

Run Super IOB 1.5 with Controller as per Issue #38 but change lines 5010 and 10010 as per above.

**11**

Enjoy program.

I would be grateful if you could print this as it is not a bad program and there are supposed to be a large number of related disks coming out (*The Dungeon*, etc).

Anyway keep up the good work and keep the Softkeys coming.

Has anyone found another protection method for the *Trillium/ Telarium* series because I have *Amazon* and *Shadowkeep*, and none of the existing methods will work on my versions.



23

Wayne Williams

First of all, I would like to say how much I enjoy reading COMPUTIST. It has been a tremendous help to me in backing up all of my software, and I think it's great that everyone is willing to share their information with other readers.

Now, since I have used all of this information, I thought I would try to contribute something.

Softkey for...

## Mindshadow

Activision

I have been able to break *Mindshadow* by Activision with a lot of help from Jeanne Edwards and her softkey for *Borrowed Time* (also by Activision) in COMPUTIST #39.

As I looked at each track with a nibble editor, I realized that the address prologues were exactly the same, and in the same track

sequence. The address epilogues also change with each sector.

Believe it or not, the same Super IOB controller that Jeanne wrote works with this program! The only differences are in the sector edits to be performed, and the locations of the prologue tables.

The procedure is as follows:

**1** Install the *Borrowed Time* controller in Super IOB, and copy both sides of the *Mindshadow* disk.

**2** Perform the following sector edits for the address prologues and epilogues.

Track	Sector	Byte(s)	From	To
\$0	\$7	\$2F-\$3E	\$xx	\$AA
\$0	\$7	\$3F-\$4E	\$xx	\$96
\$0	\$A	\$C5-\$D4	\$xx	\$AA
\$0	\$A	\$D5-\$E4	\$xx	\$96
\$0	\$1	\$FB	\$D9	\$C9
\$0	\$1	\$FC	\$2F	\$DE
\$0	\$1	\$FD	\$8A	\$EA
\$0	\$2	\$06	\$D9	\$C9
\$0	\$2	\$07	\$3F	\$AA
\$0	\$2	\$08	\$8A	\$EA
\$0	\$C	\$5B	\$D9	\$C9
\$0	\$C	\$5C	\$C5	\$DE
\$0	\$C	\$5D	\$F3	\$EA
\$0	\$C	\$66	\$D9	\$C9
\$0	\$C	\$68	\$F3	\$EA

That should do it for this one, and now it's copyable. Thanks again, and keep up the good work.



24

Dr. Dracks

## Strange April Time Slip

To start off with, I would like to ask: what happened, guys, slight time slip? When I saw LaChance's letter in COMPUTIST 42, I thought perhaps you gave pre-releases of each issue, for he was talking of an article in #42! But, then you listed bugs for #42, so, what's up guys?

Next...

a note on...

RDOS  
COMPUTIST 30, Page ??

Since Mr. Iber entered addendum to the RDOS article of issue #30, I thought I might




as well. A pity really, for I had decided to enter my method for breaking RDOS disks after #30, only to get beaten to it in #30.

Oh well. My methods are mostly different, so perhaps anyway...

Well, that article, with Mr. Iber's corrections, presents a workable method, but first of all, in it was stated that to speed up the RDOS 3.3 (deprotected) copy, one could use *Bag of Tricks* to re-init using Ascending 01. What he should have said was 11, for copies of the speed-up copy via 01 setting are slowed back down.


Also, the method of breaking into an RDOS disk is ridiculously tedious. While using 16k or 32k slave disks are fine for deprotecting some programs, why use it on RDOS disks?

## using A R D

The "ARD" disk presented awhile back is quite usable. Boot an RDOS or even RDOS 3.3 disk and hit  at a menu or even on bootup. You'll get the Applesoft prompt.

Replace the disk with the ARD and then RESET. It'll try to reboot the ARD disk.

When in the Monitor, type:

- \* C0E8 (shut drive)
- \* D6:00 (normalize RUN-flag)
- \*  (exit monitor)

Then you can list whatever Applesoft program loaded up. If I remember rightly, that is all there is to it.

## 3 cents

Further, I might as well add my three cents on the not-defunct issue over expansion of COMPUTIST.

Way to go, Mr. Boehrs! Rivett does not know what he is talking of. While IBM came on the scene first to make itself an original standard, Apple/Macs are expanding and are becoming usable in many areas.

I do not believe that Apple shall become extinct for a long time, if ever. Apple Computer have made a definite contribution to the micro-computer market and technology and is not likely to just die out.

## Mac II and //gs possibilities

Think of the possibilities of the new Mac II! The possibilities of the add-on boards that shall become available is great for the imagination. And the //gs uses with sound and graphics?! Not to mention a slot for RAM and ROM cards?

Hmmm.

## Sorry Mac

As to whether COMPUTIST should be "bilingual", I believe not. There just is not enough space. It would be nice if a sister mag, say "MacComputist" or MacAzine, could be produced by SoftKey Publishing.

My adventures into the Mac Zone hardly

rival that in the Apple ][ Dimension, but from what I see, protection is advancing, becoming widespread for Mac software, such as erased tracks, hidden files, and unstandard sector sizes, those being the most prominent of tricks.

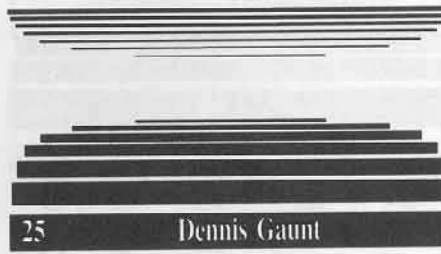
## Copy II MAC

The only sophisticated copier I have found is *Copy II MACs* (now at 6.3 version)

And as for sector editors, *Copy II MAC's* sector editor and FEdit are the only ones I know of, or even heard anything about (Both very good).

MacUsers are going to need a superior mag, like this one, to help in providing information on backing up their expensive software.

And that, for now, is it. Happy code-breaking to all.



I recently began subscribing to COMPUTIST hoping to find some way of backing up my software. None of my friends are into this sort of thing and the local user's group is little help.


My first two issues arrived and I realized I would need more help than they or the documentation given with your utilities disk could provide. I ordered and received three back issues which I thought would provide more explanation for Super IOB, DOS Alterer, and tips for beginners. Sadly, these weren't much help either.

Therefore, I am trying a variety of procedures on my own but which are based on your very informative letters from other readers. A letter pertaining to one program gave tips on similar programs.

I also wrote to you some time ago with questions but haven't received a reply. Perhaps my questions were too vague as I am still new to cracking code.

## INIT or not to INIT

For example, when using Super IOB v1.5 how are you to know whether or not to format the backup disk? Sometimes readers give this but most often not. I guess continual reading of my Apple reference books and COMPUTIST will help.

 Assume that when a blank disk is required for a softkey, it must be DOS 3.3 formatted (INITed) because IOB does not format a disk

unless the writer makes it call the INIT routine... in which case, it won't matter because it will be reINITed.

Be safe. Use an INITed disk. RDEXed

Update to Williams's Softkey for...

## Gemstone Warrior

COMPUTIST 41 page14.

Anyway, I was most interested in the deprotection of *Gemstone Warrior* by Red Williams in the COMPUTIST 41 as I had just purchased this game. It has a save-game feature which is most unhandy... it saves it back to the original disk and if any one else wants to play the game, they destroy the saved game (unless they want to continue the saved game). This simply means that multiple copies are desirable so each player can have his own saved game.

I tried Mr. Williams suggestions but they didn't work because my version of *Gemstone Warrior* has altered address epilogues as well as the altered data epilogues he mentions.

By luck and some logic (of Mr. Williams article) I was able to change some code to deprotect this issue. So, in addition to his deprotection one must also change the following:

Track	Sector	Byte(s)	From	To
00	3	91		DE
00	3	9B		AA

The altered address epilogues on my main disk are 9C and C9 rather than DE and AA.

Anyway, this works and the resultant disk is COPYable. Keep up the good work.

## Gaunt's second letter

You should have very recently received a letter from me regarding the deprotection of *Gemstone Warrior*, my version of which differs slightly from that given by Red Williams in COMPUTIST 41.

I'm afraid that in my delight and haste at finally figuring out how to deprotect a program I may have left out a step. I'm new at this and still learning.

Anyway, my version of *Gemstone Warrior* has altered address epilogues of 9C and C9 rather than DE and AA. In order to use Williams' technique it is necessary to change DOS to ignore the address epilogue.

This handy bit of information was provided by Gregory Moore in his addendum to the softkey for *The Quest* (same issue of COMPUTIST).

The full sequence for *Gemstone Warrior* with both data and address epilogues altered is as follows:

**1** In addition to patching DOS as Williams does, you need to add:

\* B988:18 60

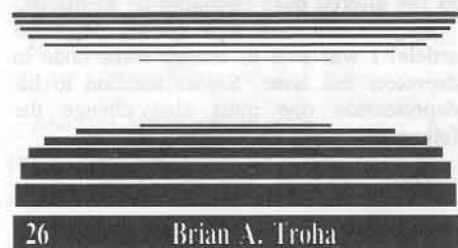
in addition to his B934:18 60.

**2** After performing the sector and byte changes suggested by Williams, you need to add the following changes:

Track	Sector	Byte(s)	From	To
00	3	91	DE	
00	3	9B	AA	

Perhaps, if you wish to use this, you could edit my letter of yesterday and this one into the coherent form. Thanks.

**I** Both letters were printed in the order they were received. We'd like to keep all editing down to a minimum. RDEXed



A note on the Sofkey for...

## Strip Poker

COMPUTIST 42, page 10

Here are a few additions I would like to pass along. First on page 10, my version of *Strip Poker* has address epilogues of "DF AA".

The softkey will still work on the main program and all the data disks.

Sofkey Update for...

## Lucifer's Realm

COMPUTIST 42, page 9

I would recommend the following sector edits (c.f. pg 10, *Lucifer's Realm*):

Track	Sector	Byte(s)	From	To
\$00	\$03	\$91	\$DF	\$DE
\$00	\$06	\$AE	\$DF	\$DE

Fast DOS addition to the Sofkey for...

## The Swordthrust Series

COMPUTIST 42 page 12

With the *Swordthrust* series if you add the following sector edit you should be able to find the files on disk with a fast DOS.

Track	Sector	Byte(s)	From	To
\$11	\$00	\$01	\$20	\$0F

A note on the Sofkey for...

## Manuscript Manager

COMPUTIST 42, page 13

Lastly, on page 13, the *Manuscript Manager*: I would recommend the following sector edits to avoid any chance of failure.

Track	Sector	Byte(s)	From	To
\$04	\$05	\$92	20	EA
		\$93	00	EA
		\$94	60	\$18

As long as the check will fail, why not avoid it all together (saving some boot time) and make the code think the check passed. This is not to say that any article was wrong, I am just providing additional information and a slightly different approach as there are always several ways to do things.

## Unlocking //gs software

For those people who plan to buy an Apple //gs, I would like to pass along this information:

**a** If you like to unprotect your software you will need to know how to boot code trace, and or use XFER.BOOT and RESTORE from COMPUTIST No. 16.

There is no way to enter the monitor, as the ROM and RAM chips are soldered in.

The *Wild Card II* hung the computer when used, I cannot say about other NMI cards.

**b** Some of the nibble-copiers that work correctly are *Copy II Plus*, *Echo 1.0*, *EDD III*, the latest version of *EDD 4*, and *Nibbles Away II*, and *Locksmith 5.0* or *6.0* ver B. The above programs were all unprotected files or disks.

**c** If your screen fills with "2" you should try setting the ALTERNATE DISPLAY MODE from the control panel. This is needed for programs that use the second text page, that is, memory from \$800 through \$BFF.

Examples include *EDD* and *Nibbles Away II*.

**d** Some protection schemes will not work and or cause the program to bomb, while an unprotected version (files) will work fine. If the protection scheme uses a time-dependent loader, chances are the program will not load. Also protection schemes that check rom locations and or checksums will fail.

Even with these problems, the Computer is a great machine. You have a vast increase in memory and double the speed. Programs like *Appleworks* will use the extra memory, while *Copy II Plus* (under ProDOS) and spell checkers benefit greatly from the RAM disk.

## New ProDOS Protection

With the new ProDOS 16 and the 3½ inch disks comes a new protection scheme. This involves having a **bad block** on a micro disk, then jumping through ProDOS for (22 A8 00 E1 22) a block-read, and finally returning the status.

One example is *Paintworks Plus*.

## Un-2-ing Pacman

On a different note...To avoid problem c above while running *Pacman* do the following:

**1** BLOAD a cracked *Pacman*

**2** Type:

J CALL -151  
\* 185F:60  
\* 1868:4C 58 FC  
\* 3F40:54

**3**

I BSAVE PACMAN, A\$1000, L\$6A00

## Improving Locksmith's Fast Backup Routine

One last thing I would like to say that the Fast Back-up routine of *locksmith 6.0* could be improved with the following edits.

This make the display easier to read and understand. The edits take and put the source and destination drive numbers from the bottom of the screen and place them next to the Read and Write letters.

Sector edit track 9 sector 8:  
Byte 07:30 8D 81 05  
Byte 10:30  
Byte 14:20 8D 01 06 A9 20  
Byte 1F:16 8D 02 06



27

Michael Cook

I've been playing *Beyond Castle Wolfenstein* for quite some time now and have enjoyed every minute of it, so here's...

## Still More Castle Wolfenstein

My current skill level is Master Saboteur which is extremely difficult because of the number of SS men and alarms in each room. If you try to shoot the SS men in the room you usually miss most of the bullets, or get caught.

### Vanishing SS

So I figured out how to kill a SS man without even shooting him or stabbing him with your dagger which I do not know how to do, if someone does know how to use the dagger, toolbox, and keys please write in and let me know.

If you have noticed, in several of the rooms there is a double wide exit in either the top of the room or the bottom, they are not hard to find. When you do find one of these exits you must maneuver one of the SS men so that he is right beside you at the exit, then leave the room.

Also, if you get him marching up and down in the right position with the exits that's fine too. Just stand there and wait for him to get in the right position then leave.

Don't worry about the SS in the room you just left. That room stays exactly the same as it was when you left.

You will probably be asked by one of the SS men in that room for your pass, do that, then go back to the exit and stand on the side of the exit the SS man was on when you left.

Then enter the room again. In other words, if you are standing on the right side of the exit, leave, go one space left, reenter the room.

There, you have just killed an SS man without even drawing your gun.

This method is extremely helpful in the alarm control room which has a spiral-like center with one door in the center and usually four SS men.

Here are some other tips.

### A Trail Of Bodies

Use this method in the room right before you get to Hitler, this room has about fifteen guards in there.

Be careful in this room, because of the large number of guards it is hard to tell which guard is talking to you. If you don't get to that guard in a certain amount of time, you're dead.

Another good idea is to leave a trail of dead

men so it is easy to find the elevator on each level or you can use the map that Lars Grant-West gave us in *COMPUTIST* No. 28 which has helped me a lot.

Here is one more hint.

**Do not shoot an SS man while he is in front of a door** or you will not be able to open the door because when you are standing in front of the door you are also standing on the dead SS. When you press the space bar to try to open the door the computer searches the dead SS, not the door.

I hope I've helped you in your quest to eliminate Hitler, good luck.

28

Jerrell Watts

a note on using...

## Exodus Construction Set

MagicSoft

A few years ago, while reading a computer magazine, I noticed an ad for the *Exodus Construction Set* in the classified section. I was quite interested (mainly because I had completed *Ultima III*, and wished to alter it). I ordered the program, and it arrived in the mail a few weeks later.

After using it for a week or two, I found it quite easy to use. I soon found myself creating whole new continents, revising cities, redesigning dungeons, and creating new and interesting characters for players to interact with.

Then came the fun part: I was able to play my modified version and see my newly created world unfold before me.

The *Exodus Construction Set* allows you to edit the following:

- A The continent of Sosaria and Lost Ambrosia
- B All of the cities and castles (including the *Exodus* castle)
- C The dungeons
- D The positions of the monsters and town folk, as well as what they say and do.

For instance, using *Exodus Construction Set* you could edit Sosaria and have the city of Yew surrounded by water.

Then you might edit Yew itself and create a new shop, say, an oracle. Then you might create a merchant to run it and a wizard customer who says a few words of praise

regarding the oracle.

Then you could edit a dungeon and redesign the first three levels to suit yourself.

## More From MagicSoft

About a month later, I received a new product notice from MagicSoft introducing two new programs, *Wizi-Doc* (a Wizardry character editor) and *Wizi-Scout*.

I was particularly interested in *Wizi-Doc* because I had recently lost one of my best characters while adventuring. Both programs arrived within a few weeks after I ordered them.

*Wizi-Doc* allows you to do the following:

- A Edit all attributes (strength, agility, luck, etc.)
- B Edit hit points, gold, experience points, level, alignment, class, etc.
- C Give your characters new items and uncurse those items they have
- D Give your characters more spells

I used *Wizi-Doc* to revive my dead character, increase his strength, and uncurse a Deadly Ring I had found.

*Wizi-Scout* allows you to find the powers, abilities, resistances, and weaknesses of any monster in the *Wizardry* series. It also allows you to discover the powers of any magic item. I used it to find the abilities of the Vampire Lord, and to discover the magic powers of the Rod of Flame.

## Some Drawbacks

Of course, as with any program, there are weaknesses in these MagicSoft programs.

*Exodus Construction Set* lacks the ability to relocate cities, castles, dungeons, and moon gates, which means you have to design your world to fit their locations.

*Wizi-Doc* lacks slightly in documentation and its editing mode is somewhat crude.

*Wizi-Scout* has no documentation, but in most cases this lack is not noticed as the program is quite self-explanatory.

All in all, These MagicSoft programs are excellent, and despite their few limitations are well worth the money. I wholeheartedly recommend MagicSoft products to any player of adventure games.

## Still More Products

MagicSoft has recently released several new programs:

*Adventure Scout vols. 1 & 2* (\$19.98 each) - hints, tips, and maps for many popular adventure games

*Atomic Wasteland 2098* (\$29.99) - a *Wizardry* scenario set in the future

*Bard's Tale Doc* (\$19.99) - a *Bard's Tale* character editor

*Ulti-Doc Plus* (\$24.99) - an *Ultima III* and *Ultima IV* character editor

*Ultima IV Construction Set* (\$39.99) - edit many aspects of *Ultima IV* as with *Exodus Construction Set*

*Wizards Workbench* (\$34.99) - create or modify *Wizardry* scenarios

Enjoy!! I hope this tip might be of some use to other gamers in the APT hunt.

30 Eric W. Taylor

I recently bought *Bank Street Writer III* from Scholastic. I am very impressed with the program and I use it all the time. It has nice mouse capabilities and is very user friendly.

The only problem with it is the fact that it is copy protected. I set out to see what I could do to remedy the situation.

Softkey for...

## Bank Street Writer III

Scholastic

After a couple of hours of snooping, I came to the conclusion that I did not have enough time in my schedule to figure out a softkey for it.

The format of the disk is very strange. The headers and trailers are messed up. One interesting feature of the original was the fact that track zero was almost normal.

I took this as a good sign. I also discovered that some of the information on track zero was encoded. Specifically, the second stage of the boot. By tracing the boot process and writing some assembly code, I figured out what the program was reading in off of the disk. If these values were not the right ones, the disk re-booted.

The rest of the tracks were not at all normal, but they looked as if they could be nibble copied. Once I nibble copied the disk and applied my patch to track zero, the disk booted fine. The procedure is not very hard and I figure as long as I have a backup, I do not have to worry about my original. The procedure outlined below in cookbook form worked for me and for friends of mine who have *BSW III* originals. Hope it works for you.

**1** Use *Locksmith 6.0* normal copy to nibble-copy the original. Copy the whole disk from track zero to track \$22.

**2** Using *Tricky Dick* or some other sector-editor that can read in abnormal data epilogues, read in track zero, sector \$A from the original using a data epilogue of \$DE BF.

Write this sector back to the copy using the normal epilogues of \$DE AA. You can now lock your original *BSW III* in the vault along with your other originals.

**3** Read in track zero, sector \$9 from the copy. Make the changes in Table 1 starting at byte \$57. This patch forces the *BSW* disk to think the nibble count has been satisfied and allows the boot to continue.

Table 1

TRACK	SECTOR	BYTE	FROM	TO
\$0	\$9	57	BD	A9
\$0	\$9	58	8D	FC
\$0	\$9	59	C0	85
\$0	\$9	5A	A0	F0
\$0	\$9	5B	10	85
\$0	\$9	5C	24	F3
\$0	\$9	5D	80	85
\$0	\$9	5E	BD	F6
\$0	\$9	5F	8C	A9
\$0	\$9	60	C0	EE
\$0	\$9	61	10	85
\$0	\$9	62	FB	F1
\$0	\$9	63	88	85
\$0	\$9	64	F0	F2
\$0	\$9	65	26	85
\$0	\$9	66	C9	F5
\$0	\$9	67	EE	A9
\$0	\$9	68	D0	E7
\$0	\$9	69	F4	85
\$0	\$9	6A	EA	F4
\$0	\$9	6B	EA	85
\$0	\$9	6C	A0	F7
\$0	\$9	6D	07	EA
\$0	\$9	6E	BD	EA
\$0	\$9	6F	8C	EA
\$0	\$9	70	C0	EA
\$0	\$9	71	10	EA
\$0	\$9	72	FB	EA
\$0	\$9	73	99	EA
\$0	\$9	74	F0	EA
\$0	\$9	75	00	EA
\$0	\$9	76	EA	EA
\$0	\$9	77	88	EA
\$0	\$9	78	10	EA
\$0	\$9	79	F4	EA

**4** That's it. Write the patch back to track zero, sector \$9. The copy should now work nicely.

The main program should work and the utilities section that is reached by pressing the escape key during the boot should also still work. This procedure has worked on the eighty column, forty column, twenty column and the *ProDOS Utilities* disks.

The speller, thesaurus and tutorial disks may all be fast-copied. Have fun.

29 Jim S. Hart

Lost Tomb is one tough game. Veterans of the game know what I am talking about. Beginners rarely venture beyond the first two levels for quite some time.

After playing the game for a couple of weeks and getting nowhere fast, I figured it was time to 'explore' the pyramid. This meant it was time to get unlimited men and whips.

A P T for...

## Lost Tomb

In most games, different memory locations hold items of data such as the number of men left, ship damage, etc. When a man is lost, that location is decremented by one.

My idea was to scan through the *Lost Tomb* code for the 'decrement memory location by one' opcode (\$CE). I wrote down where each of them occurred, proceeded to change them one by one, and observed what happened after the change was made.

It turns out that the 'men' memory location was at \$2F4 and the 'whips' location was at \$2F1.

## Unlimited Men and Whips

To get unlimited men, scan your *Lost Tomb* disk or file for the bytes CE F4 02 and replace them with EA EA EA.

The whips location was a bit trickier to find. The whips number is loaded into the X-register and then the X-register is decremented. This is then checked to see if it is positive (more whips left to use) and put back into the memory location if so. Luckily this code is right near the 'men' code or it might have never been found. To get the ability to have unlimited whips, search the *Lost Tomb* code for the byte sequence AE F1 02 F0 0A CA and change the final byte (CA) to a EA. This change should be pretty close to the unlimited men change.

31

Marc Batchelor

*Kids On Keys* (KOK) is a very basic typing tutor program aimed at young children. It has 3 different typing games and up to 4 levels. Points are earned by typing in the correct corresponding letters before they fall out of sight, or identifying and spelling the names of crude drawings before they fall off the screen.

All in all, KOK is average in comparison with similar programs on the market (*Type Attack*, *Masterytype* etc.).

Now, on to the...

### Softkey for...

## Kids On Keys

Spinnaker Software

#### ■ Requirements:

- Whole-disk copier
- Sector Editor

I started by trying the controllers listed in previous magazines. I tried the controller for *Grandma's House* and *Alphabet Zoo*, also from Spinnaker. All Super IOB did was grind.

I then decided to look at the disk with my trusty nibble editor. I quickly noticed that the disk was in standard DOS 3.3 format.

The controllers were trying to read a disk with a format other than normal. So, I copied the disk with the Copy Disk option of *Copy II Plus* and noticed that tracks \$1F — \$22 were **unformatted**.

Booting the copy disk, I thought maybe it wasn't protected to start with. No dice!

As it turned out, a nibble count was being performed. I pressed reset and the disk re-booted.

Well, I decided to search the disk for the bytes F2 03, to see if they fooled around with the Reset vector.

To my surprise, I found about 7 places that changed the Reset vector. After changing them all to 69 FF 5A's, I booted the disk and when I received the "Please Stand By for Kids On Keys" message, I pressed Reset.

As expected, I was put in the monitor. I then went to BASIC, and tried a LIST. Amazingly enough, a BASIC program appeared. I was even able to get a CATALOG by typing "CALL 42350".

Unfortunately, I still hadn't located the dread nibble count. What now?

Well, I got back to the point where I was in

the monitor just after the startup message. I typed "MONCIO" to follow what files were loaded when I noticed right after the KICK file was loaded, memory was bombed.

With a bit of tracing through this file, I found a JMP \$888F that seemed not quite right. I changed it to a JMP \$4000 and viola, the disk worked perfectly.

### Step by Step

**1** Copy the disk ignoring the errors on tracks \$1F-\$22.

**2** Use a sector-editor to read track 14, sector \$0E.

**3** Change bytes 9D and 9E from 8F 88 to 00 40.

**4** Write the sector back to the disk.

That it. Now here's a....

### Softkey for...

## The Missing Ring

Datamost

#### ■ Requirements:

- Blank Disk
- File copier

*The Missing Ring* is a poor rendition of *Ultima*.

The biggest differences between the two in that in *Ultima*, you can move "the party" at one time, and are on the individual level during battle, while in *The Missing Ring*, you have to always move the individuals one at a time, battle or not.

The other big difference is that *Ultima* has nice graphics. The *Missing Ring* however does not.

At first, I was not aware that the disk was protected. I copied it with *Locksmith Fast Copy*, and it booted fine. I could even CATALOG the disk from normal DOS.

### Landing In Nowhere Land

Although no check was performed during boot-up, when a member of your party decides to take some treasure, the disk drive comes on, there are a few clicks, and then the screen fills up with several letter "D's, and then the computer is off in no-where-land.

Pressing Reset at this point causes a total re-boot.

Now, I decided to see exactly how open this

disk really is, so, I booted the disk and immediately pressed [C].

Sure enough, the disk crashed into BASIC.

Now, I tried a CATALOG... beep SYNTAX ERROR.

So, I typed "CALL 42350". Sure enough, a valid CATALOG appeared.

Then, I tried to LOAD a file, just to see if that command worked. I was able to load files, so I quickly dragged out my copy of *DOS Boss* (Beagle Bros.) and put it in the drive.

I was able to load and run *DOS Boss* to find out that CATALOG had been changed to CAT. I tried CAT, and amazingly enough, it worked too.

### Looking For Treasure

Realizing that this program started having problems when treasure was taken, I noticed a program called TREASURE in the directory.

First, I loaded another program on my shelf called *PLE*, or *Program Line Editor* and started flipping through the program.

Five hours later, I realized that this program was not the problem.

Instead, it must be in the program that called this one.

After flipping through the directory again, I noticed a program called PROG2. So, I first dug into my Beagle Brothers utilities, and came up with a program called D-CODE. This program lets me do incredible things with my Applesoft programs. Anyway, I B-ran the file D.BUG and then put *The Missing Ring* disk back into the drive to load up PROG2. After loading it up, I typed "FL" for "Find List" or find what I'm looking for and LIST the line on which it occurs. When asked what to search for, I typed "TREASURE". A few seconds later, D.BUG listed line 9280. The line read:

```
9280 IN$ = "TREASURE" : GOTO 9500
```

Going to line 9500, I found this:

```
9500 GOSUB 450 : PRINT CHR$(13) ; CHR$(4) ;  
"BRUN READ.WRITE" : PRINT CHR$(13) ;  
CHR$(4) ; "RUN" ; IN$
```

I therefore changed line 9500 to:

```
9500 PRINT CHR$(13) + CHR$(4) ; "RUN" ; IN$
```

...and wrote the file back to the disk. *The Missing Ring* then functioned perfectly.

### Step by Step

**1** Initialize a blank disk (preferably with a fast DOS).

#### I INIT HELLO

**2** Copy all files off *The Missing Ring* disk to the disk you just INITed.



**3** Insert the copy disk, load the PROG2 file and change it.

**J LOAD PROG2**

**J 9500 PRINT CHR\$(13) + CHR\$(4) ; "RUN" ; IN\$**

**4** Save the program back to disk.

**J SAVE PROG2**

**5** Put the original disk away in a safe place.

**32** Christopher James Dean

The *Graphic Solution* (TGS) is a graphics utility that helps you make movies.

You can draw Lo-Res shapes and place them on the Hi-Res screen to make backgrounds and scenes. *TGS* also has a circle function and a projector program that you can use with your own programs to display your movies.

Softkey for...

*Graphic Solution*

Accent software

**■ Requirements:**

- Apple ][ plus, //e or //
- Super IOB
- A Sector Editor
- 1 blank disk side

## The Protection

1. *TGS* is written in a normal DOS 3.3 format with **invalid data checksums**. The checksums are off because the disk is encoded.

2. The RWTS (Read and Write translate tables are in reverse order. Thus, the disk cannot be read correctly with a normal RWTS.

3. The disk also contains a modified DOS.

The programs themselves are protected slightly.

4. Four of the programs set the RUN flag by poking 214,255, so that if the program is broken into and a command typed, it will run the program.

5. Several of the programs also use the INIT command, which on the master disk has been nullified.

6. Several of the lines in the programs are also too long to edit with a normal DOS so the INIT function must be nullified on the copy also.

## The Procedure

**1** Boot normal DOS and initialize a blank disk with TGS48 as the hello program.

**J INIT TGS48**

**2** Now we must use a RWTS Swap controller to copy *TGS*. To do this we must create a RWTS that will read *TGS*.

**J CALL -151**

\* 6800<B800.BFFFM

\* 6A96: 3F 3E

\* 6A98: A7 A6 3D 3C A3 3B 3A 39

\* 6AA0: 9F 9E 9D 9C 9B 9A 38 37

\* 6AA8: 97 96 95 36 35 34 33 32

\* 6AB0: 8F 8E 31 30 2F 2E 2D 2C

\* 6AB8: 87 2B 2A 29 28 27 26 25

\* 6AC0: FF FE FD FC FB FA F9 F8

\* 6AC8: F7 F6 F5 24 F3 23 22 21

\* 6AD0: EF EE ED 20 EB EA 1F 1E

\* 6AD8: E7 1D 1C 1B 1A 19 18 17

\* 6AE0: DF DE DD DC DB 16 15 14

\* 6AE8: D7 13 12 11 10 0F 0E 0D

\* 6AF0: CF CE 0C 0B 0A 09 08 07

\* 6AF8: C7 06 05 04 03 02 01 00

\* BSAVE SOLUTION.RWTS, A\$6800, L\$800

**3** Next, enter the swap controller into Super IOB and copy *TGS*.

**4** Using a sector editor, edit track \$01, sector \$04 and change byte \$4F from an \$A9 to a \$60. This will make the INIT function useless.

**5** Finally, make the following changes to these files to ignore setting the Run flag:

**J LOAD MENU.48**

**J 3 HOME : VTAB 2 : HTAB 11 :  
INVERSE : PRINT "TGSΔ MENUΔ  
SELECTION" : NORMAL**

**J SAVE MENU.48**

**J LOAD MENU.4864**

**J 3 HOME : VTAB 2 : HTAB 11 :  
INVERSE : PRINT "TGSΔ MENUΔ  
SELECTION" : NORMAL**

**J SAVE MENU.4864**

**J LOAD TGS64**

**J 12**

**J SAVE TGS64**

**J LOAD TGS48**

**J 12**

**J SAVE TGS48**

Happy Movie Making!

**33** Charles S. Taylor

Softkey for...

*Empire I*

and...

*Empire II*

**■ Requirements:**

- A way into the monitor

## The Protection

These are the first two components of the *Empire* trilogy, a science fiction role-playing game.

They are copy-protected by schemes familiar to most COMPUTIST readers. *Empire I* has an altered RWTS, while *Empire II* is protected by non-standard address and data epilogues.

## The Procedures

**1** Boot the DOS system master and initialize a blank disk.

**J INIT INTERACTIVE FANTASIES**

**2** If your way into the monitor is by using XFER.BOOT and RESTORE (from COMPUTIST 16), then BLOAD it now.

**J LOAD XFER.BOOT**

**3** Put your *Empire I* disk in the drive and boot it.

A If you are using the XFER-RESTORE method type:

**1 CALL 768**

B Otherwise, use:

**1 PR#6**

**4** Reset into the monitor. If you are using the XFER-RESTORE method, just press Reset and you will be in BASIC.

**5** Move the Empire RWTS to a safe place in memory.

A If you are using the XFER-RESTORE method, use:

**1 CALL -151**

\* 3F8:4C 00 03

\* BLOAD RESTORE

\* 1900 < B800 .BFFF  Y

B Otherwise, use the following command and then boot a disk with no hello program.

\* 1900 < B800 .BFFF M

\*  P

**6** Save the modified RWTS.

**1 BSAVE RWTS.EMPIREΔI, A\$1900, L\$800**

**7** Merge the Swap controller with Super IOB and use the resulting program to copy EMPIRE I.

Make sure that line 10010 BLOADs the RWTS file you just saved.

## Now For Empire II

**1** Boot the system master and modify DOS so it can read the *Empire II* disk.

**1 POKE 47426, 24**

**2** Now, use COPYA to copy the disk.

**1 RUN COPYA**

**3** With a sector editor, change the DA on Track 0, Sector 2, byte 9E and Track 0, sector 3, bytes 35 and 91 all to a DE.

34 Jim S. Hart

## Fool the Graphics Expander!

### How To Convert... *Newsroom Clip Art* into *Print Shop Graphics Expander* Disks

**What you need for this venture:**

- Deprotected *Newsroom Clip Art* disk(s)
- Deprotected *Graphics Expander* disk
- A blank disk for each *Clip Art* disk you have
- A sector-editor
- Print Shop* (why else would you have the *Graphics Expander*?)

The *Print Shop* is a very handy program to have. You can print out slick looking banners, signs, and greeting cards, among other things. Its success has spawned several programs that give you extra graphics to use along with it.

At last count there were three *Graphics Library* disks, the *Graphics Expander*, an extra graphics disk from the folks who put out *Print Master*, and many other disks that have more graphics to use with the *Print Shop*.

Here is my method to give *Newsroom Clip Art* and *Print Shop Graphics Expander* owners even MORE graphics.

### What I found

While looking through the *Graphics Expander*, I noticed that the graphics it has look an awful lot like the graphics from the *Clip Art A & B* disk that comes along with the *Newsroom*.

On a hunch, I inserted a deprotected *Clip Art* disk into the drive when prompted to put in the back side of the *Graphics Expander* disk. No luck.

I then re-read Doni Grande's article on *Newsroom* (COMPUTIST 23) and found where the program looks to find out which disk is in the drive.

The first eight bytes of track \$22, sector \$0 contain this information.

Here are the identification bytes for each type of disk:

Type of disk	disk ID
Clip Art disk	SSI CLIP
Graphics Expander disk	SSI GECL

## Changing The ID Bytes?

I wondered. What if I made a dummy copy of the *Clip Art* disk, changed the ID bytes to make it look like the *Graphics Expander* disk, and then used it in place of that disk?

Well, I'm happy to say that it worked like a champ! I now can use *Clip Art* disks for my *Print Shop* graphics needs (after first using the *Graphics Expander* to convert them, of course).

There is only one limitation, however. You cannot use large *Clip Art* graphics due to the fact that the *Graphics Expander* was not set up to use large graphics. If you try, your computer will go into never-never land and not return for a LONG time.

This is not that big a limitation, though.

Here are the instructions for converting a *Clip Art* disk into a *Graphics Expander*-readable disk below.

You could even reverse the process and use *Print Shop* graphics in your *Newsroom*!

### Step by Step

**1** The first thing to do is copy your deprotected *Clip Art* disk(s) onto the blank disks. These will be your work disks to use with the *Graphics Expander*.

**2** Put away your original deprotected disks now since they are no longer needed.

**3** Boot up your sector-editor.

**4** Place one of the *Clip Art* copies into the drive and read in track \$22, sector \$00.

**5** Change the first 8 bytes so that they read "SSI GECL" (in text).

**6** Write this sector back to the disk.

**7** Repeat steps 4-6 for each work disk that you made in step 1.

**8** You may now use these disks in place of the *Graphics Expander* graphics disk. Enjoy the extra graphics!

35

Mike Newlicp

Softkey for...

## Championship Golf

Hayden

### Requirements:

- Super IOB
- FID or other file copier

OK, you golfers, here's a game to test your skills. You can choose up to 20 different clubs and you can also use iron wedges along with all the other goodies golfers use.

Ah, but since the company knew it's such a great game, it protected it. But never fear! Super IOB is here.

## The Protection

The disk uses an altered DOS 3.2 format as its form of protection. When one sees a "J" when the disk boots, there is a 100% chance that the disk uses an altered DOS 3.2 or 3.3 format.

Using the *Controller Writer* (COMPUTIST 16), I was able to make a controller for this disk after I had captured its RWTS.

After trying my controller, I observed that the last three sectors of each track were giving me errors so I concluded that the disk was in a thirteen sector format.

I wrote another controller to suit the thirteen sector format, and then it copied great!

To hinder the would-be copied, the company didn't bother to format track five, sector zero to track six, sector zero. By using the "ignore read errors" in the *Controller Writer*, I was able to defeat this.

- 1** Initialize a blank disk.

### 1 INIT GOLF

**2** Boot up the original. When the "J" appears, keep on hitting the reset key until you get a prompt. After you get the prompt, enter the monitor and move the RWTS to a safe location so it won't be destroyed when the disk is rebooted.

### 1 CALL-151

\* 4800<B800.BFFFM

**3** Insert the newly initialized disk and reboot.

\* C600G

**4** When the drive stops running, save the foreign RWTS to disk.

**1** BSAVE CHAMP.GOLF.RWTS,  
A\$4800, L\$800

**5** Copy this file using FID or some other program to your Super IOB disk. Now type the controller in at the end of the article, run it, and now you're almost finished.

**6** Once you've run the controller, load in the file called "GOLF" and make one minor addition to it.

### 1 UNLOCK GOLF

### 1 LOAD GOLF

**1** 900 CALL -3100 : POKE 49234,0 :  
?CHR\$(13) ; CHR\$(4) ; "BLOAD  
HAYPAGE" : WAIT -16384 ,128 :  
TEXT

### 1 SAVE GOLF

### 1 LOCK GOLF

**7** Now you have a COPYAable *Championship Golf*.

## Controller

```
1000 REM CHAMPIONSHIP GOLF CONTROLLER
1010 TK = 0 : LT = 35 : CD = WR : MB = 130 : ONERR
      GOTO 550
1020 ST = 0 : T1 = TK : GOSUB 490 : GOSUB 360
1030 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST
      < 13 THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 : TK = TK + 1 + (TK = 4) : IF TK < LT
      THEN 1030
1060 GOSUB 360 : TK = T1 : ST = 0 : GOSUB 490
1070 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST
      < 13 THEN 1070
1080 ST = 0 : TK = TK + 1 + (TK = 4) : IF BF = 0
      AND TK < LT THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : AS = "ALLADONE" : GOSUB 450 : END
10010 PRINT CHR$(4) "BLOAD CHAMP.GOLF.RWTS
      , A$1900"
```

## Controller checksums

1000 - \$356B	1060 - \$4AE0
1010 - \$CBEE	1070 - \$A480
1020 - \$8906	1080 - \$06B4
1030 - \$489F	1090 - \$07B8
1040 - \$4C52	1100 - \$0108
1050 - \$8917	10010 - \$3F04

## Most Wanted List

### Do You Need Help Backing-up A Particularly Stubborn Commercial Disk?

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

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

If you know how to deprotect unlock, or modify any of the programs below, let us print your know-how in the *Readers Data Exchange* soon. Send the information to us in a letter to the RDEX editor preferably on a DOS 3.3 diskette.

Apple Business Graphics Apple Computer  
Visiblend Microlab  
Gutenberg Jr. & Sr. Micromation LTD  
Prime Plotter Primesoft Corp.  
The Handlers Silicon Valley Systems  
The Apple's Core: Parts 1-3 The Professor  
Fun Bunch Unicorn  
Willy Byte ... Data Trek  
Snoggle Broderbund  
ABM Muse  
Agent U.S.A. Scholastic  
Handicapping System Sports Judge  
Odin Odesta  
Mabel's Mansion Datamost  
Brain Bank The Obsevatory  
Crypt of Media Sir Tech  
The Works First Star Software  
Cross Clues Science Research  
Peeping Tom Microlab  
Jigsaw Microfun  
Miner 2049er II Microfun  
Create with Garfield DLM  
Print Master Unision World  
Bandits Sirius Software  
Bank Street Filer Broderbund  
Operation Frog Scholastic Software  
Work Force II Core Concepts  
Super Boulder Dash Electronic Arts  
Zorro Datasoft  
Goonies Datasoft  
Kitchen's Game Maker Activision





# How To Interrupt Your Apple

by Bobby

Recently, while flipping through an old issue of CALL APPLE, I ran across a hardware article that showed how to build a card to generate interrupts. The interrupts were at a preset interval of 60 times/sec and could be switched on or off by software. That's when I made big mistake number one. I loudly exclaimed to the other computists, "What a stupid idea. Of what possible use are interrupts on an Apple II?"

I was promptly assailed on all sides. Apparently, you can do a lot with interrupts.

Then I made mistake number two. (Open mouth, insert foot, chew vigorously) I said, "Well it's still a stupid card. It only has one interval and it costs thirty (\$30) dollars. I can build a better one for less and it would be programmable and monitor real world events." The answer was unanimous. They said, "Oh yeah! So do it."

When you paint yourself into a corner, the only sensible thing to do is to admit you were wrong. Naturally, I answered, "I will." (Keep on chewing)

Several evenings and lots of paper later, I began to realize the size of the task I had let myself in for. Designing a fully programmable delay circuit with real world inputs was more than just a week end project. So I gave up on the idea of using simple IC chips to make a card and started looking at my data books for something close to what I needed that I could modify and make do.

Murphy must have been out of town or my fairy godmother was passing by because I found what I needed. Not just close, but exactly what I needed.

## The 6522 VIA Versatile Interface Adapter

This 40 pin chip has two 8-bit programmable ports, four programmable control lines, two 16-bit programmable counters, a shift register and is set up specifically to generate interrupts. The operative word is programmable and the chip only costs \$3, a long way from \$30.

There are 16 registers that control all the functions of the VIA. I will explain only those registers that affect the two timers and the four input lines.

### Controlling the Interrupts

All interrupting is controlled by the Interrupt Flag Register (IFR) and the Interrupt Enable Register (IER) in the following manner. The two timers and the four inputs can each set

(bit=1) a bit in the IFR. If the corresponding bit in the IER is set, then the IRQ output will go low. If the IER bit is clear (bit=0) then the potential interrupt is ignored and the IRQ output stays high.

Individual bits in the IFR can be cleared by writing to the register (\$C0nD). A one in the data word will clear that bit, a zero will leave it unchanged. You cannot set bits by writing to the IFR. Bit 7 of the IFR is high whenever the IRQ output is low and vice-versa. This bit is unaffected by writes to the IFR.

Individual bits in the IER can be set or cleared by writing to the register (\$C0nE). Bit 7 determines whether you are setting or clearing. 0 = clear. 1 = set. Again, when writing to the register, a one bit will cause a change and a zero bit is ignored.

Reading either register does not affect the contents of the register. Remember, bit7 of the IFR is high if the chip is generating an interrupt and bit7 of the IER is always low for a read.

Bit	7	6	5	4	3	2	1	0
IFR	IRQ	T1	T2	CB1	CB2	SR	CA1	CA2
IER	Set/Clr	T1	T2	CB1	CB2	SR	CA1	CA2

### Real world inputs

There are four edge triggered control lines (CA1, CA2, CB1 and CB2). The polarity of the edge is programmed by the Peripheral Control Register (PCR). A write to \$C0nC will directly affect the data in this register.

Bit	7	6	5	4	3	2	1	0
PCR		CB2	CB1	CA2	CA1			

Bit 0 controls the polarity of the active transition for CA1. 0 = negative edge, 1 = positive edge.

Bits 1-3 control the polarity of the active transition for CA2. 001 = negative edge, 011 = positive edge.

Bit 4 controls the polarity of the active transition for CB1. 0 = negative edge, 1 = positive edge.

Bits 5-7 control the polarity of the active transition for CB2. 001 = negative edge, 011 = positive edge.

### Timer 1

Timer 1 is a 16 bit down counter (You load it and it counts down to zero) with two 8 bit latches. It uses Phase0 as a clock. There are four registers that affect operation of this timer (\$C0n4-\$C0n7). They operate as follows;

\$C0n4 - Write to lower 8 bit latch - read lower

8 bits of counter, reset T1 interrupt flag.

\$C0n5 - Write to upper 8 bit latch, write to upper 8 bits of counter, transfer lower 8 bit latch contents to lower 8 bits of counter, reset T1 interrupt flag (IFR bit6). - Read upper 8 bits of counter.

\$C0n6 - Write to lower 8 bit latch. - Read lower 8 bit latch.

\$C0n7 - Write to upper 8 bit latch, reset T1 interrupt flag. - Read upper 8 bit latch.

### Timer 2

Timer 2 is an 16 bit down counter with one 8 bit latch. It uses Phase0 or PB6 as a clock. There are two registers that affect operation of this timer (\$C0n8-\$C0n9). They operate as follows;

\$C0n8 - Write to lower 8 bit latch - read lower 8 bits of counter, reset T1 interrupt flag.

\$C0n9 - Write to upper 8 bits of counter, transfer lower 8 bit latch contents to lower 8 bits of counter, reset T1 interrupt flag (IFR bit6). - Read upper 8 bits of counter.

### Controlling Timer Modes

The Auxiliary Control Register (ACR) sets the operating mode for both timers. A write to \$C0nB will directly change the data in this register.

Bit	7	6	5	4	3	2	1	0
ACR		T1	T2					

Timer 1 can operate in one of four modes determined by ACR bits 7 and 6.

00 = generate one interrupt for each load  
01 = generate continuous interrupts

10 = generate one interrupt for each load and a pulse on PB7.

11 = generate continuous interrupts and toggle PB7.

Timer 2 can operate in one of two modes determined by ACR bit 5.

0 = generate one interrupt for each load, continues to decrement.

1 = generate one interrupt for each load, uses PB6 as a clock, continues to decrement.

### What's it all mean?

Let's take another look at what we've got. We have four input lines (five if you count PB6) which we can connect to various points in the Apple that we may want to monitor. We can select whether we want to have a negative or positive edge trigger on each of the four inputs. So, what if we connected the paddle buttons to



two of the inputs? Then we wouldn't have to monitor the buttons, just let the interrupt tell us when someone has pressed a button. Or we could connect an input to the keyboard AKD pin and forget about checking the keyboard. Or how about connecting PB6 (negative edge triggered) to the WNDW signal (output enable on the video ROM). The WNDW signal consists of 192 negative pulses (40 microseconds low, 25 microseconds high) followed by a long high of 4550 microseconds. Recognize that? 192 lines or  $192/8 = 24$  lines. Right, the hi-res or text screen. The long high lever is the vertical sync. We could poke a number into Timer 2 and when the video scan hit that line, instant interrupt. Think of the special effects. We could program Timer 2 to generate an interrupt for each vertical sync or for every third or every tenth. Then the interrupt handling routine would move things around on the screen. Background animation, handled automatically. This is kind of neat.

Timer 1 can be used to generate delays from microseconds to milliseconds. Program the correct interval and count it at each interrupt and you have a software clock. Or write a printer buffer that stores data in memory. Then uses the timer to generate interrupts. At each interrupt, sent some characters to the printer interface and instruct the timer to call you again later, if there are more characters to print. And then there's the PB7 output. We can use this output to generate a square wave with selectable period. And, since it is possible to write to the latches without affecting the contents of the counter or the count down in progress, we could continuously change the contents of the latches and generate any sort of complex waveform. You know, make weird noises.

How about making the output of T1 (PB7) into the input of T2 (PB6) for some really long delays. From microseconds to hours.

All this from one \$3 chip and we aren't even using all its functions. See, I told you it would be easy.

### Making the card

Now let's take a look at what it takes to make this super interrupt card. We need one 6522 VIA chip, one bidirectional buffer (74LS245), maybe an open collector buffer (74LS09) just to be sure and a few square inches of circuit board. Hmm, don't seem to bad. The schematic is all you really need to kludge everything together. An Apple plug-in prototype board with solder pads and power supply lines costs about \$25. The 74LS245 and the 74LS09 should run you about \$1.50 for both. And the VIA is \$3. That comes to \$29.50 and that's too close to our target figure of \$30.

So here's the bottom line. I called around and found a place that will make the boards for \$15 each but I have to order at least 100. That means the total cost of the card, assembled, is less than the cost of just a prototype board. That is, if there are at least 100 of us who want this card enough to pay \$15.

Here's what you do. Get your checkbook and write a check for \$17 to COMPUTIST (\$15 for the card and \$2 for shipping).

Send it to COMPUTIST, Attn Interrupt Card, PO box 110846, Tacoma, WA 98411-0846. They'll stack your letters in a box and if they get 100 responses, order the cards. Otherwise, your letters will be returned, unopened. So don't send anything else in the envelope except the check and your order.

### Other cards

While I've got your attention, there are a few other things pertaining to hardware that I need to know. I have a few other ideas for nifty cards and I want to know how many of you would be interested. The costs are only guesstimates (since only the Instant On card is finished) and don't include the cost of assembly. These are kit prices and include all parts plus circuit board. The cost figure is included to help you gauge your interest level. The actual price could be higher or lower depending on how many cards we actually make.

1. The **TEXT** card. (\$45) This card would allow you to replace the Video ROM with RAM that maps into the \$C800 area. This means that you have instant programmable character sets that will work with you other software without modification. Your text editor for example. It also allows graphics and animation on the text screen in monochrome. Sorry, this one only works on an Apple IIe or compatible with the same Video ROM setup. The Video ROM must be socketed and use a 2732 ROM.

2. The **INSTANT ON** card. (\$30 with one ROM) This card has space for 2-64K ROMs and 1-32K RAM. It is activated by reset. It can be used to load DOS or other programs into your Apple in an instant, hence the name. It can also be used to store utilities and other programs that you use a lot or just like to have around. I suppose I could make it activate by push button and generate an NMI, but that would make it into a cracking card. You decide.

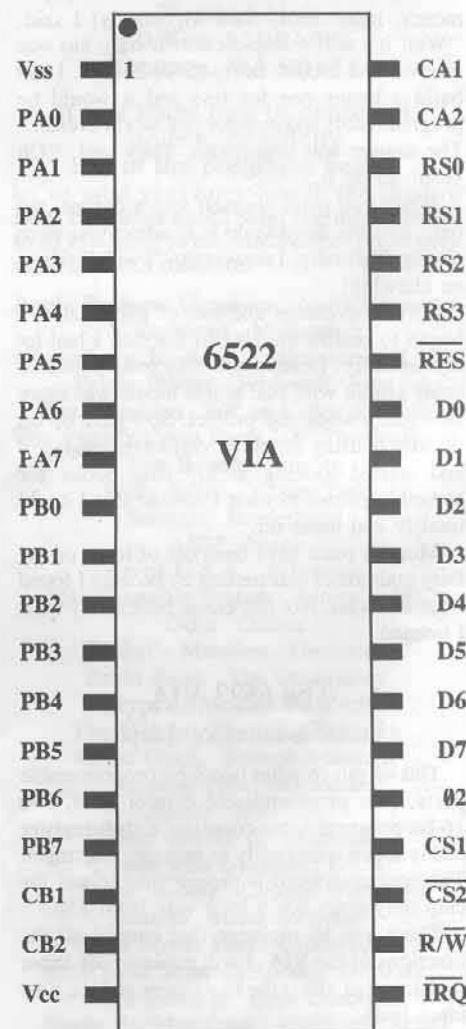
3. The **DISPLAY** card. (\$35) This one monitors the upper 4 bits of the address bus and displays the location as 1 of 16 lights (LEDs). This tells you where a program is and where it is referencing. The intensity of each light indicates how much time is being spent in each block. An option to use the sync line would cause the display to only show where the program is residing. Just for grins, I can put in a circuit to slow or single-step the Apple. It would make it easier to trace a running program. It could also monitor and display the status of certain soft switches so you could know at a glance what screen is being displayed and whether you are in main or aux memory.

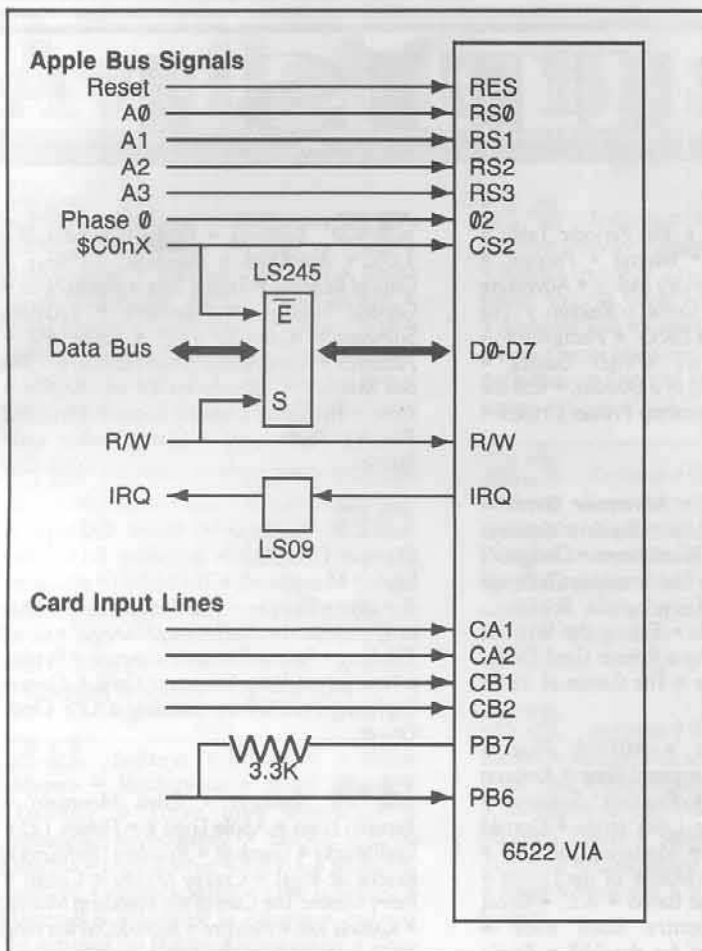
4. The **COMPUTIST Cracking Card**. (\$100) This card comes with its own RAM, ROM and processor. It can go anywhere in the Apple, read or write and exit without leaving any traces. It can generate IRQ, NMI and RDY signals to interrupt (there's that word again) or slow and stop the Apple. It comes with a 6522 VIA on board. RAM can be expanded to 128K and ROM to 128K. A piggyback board allows another 128K of RAM. It uses a 65816 processor with an asymmetrical and asynchronous clock and runs at an effective speed of 4 Mhz.

Half of that when in ROM. It can actually move blocks of data, in the Apple, two and a half (2.5) times faster than the Apple itself can. Makes for some rather interesting graphics possibilities.

### Please write

Write to COMPUTIST, RDEX Editor and let them know what you think. They'll let me know what interest there is and what card to concentrate on. If you have any other ideas for hardware, let them know that to. I really want to build these cards but COMPUTIST won't pay for the parts unless enough of you readers indicate an interest. Let's make Hardware Corner a regular column.





**APPLE COMPATIBLES  
LOWEST PRICES ANYWHERE!**

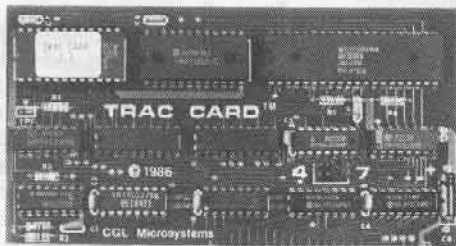
64/80 Column BD (Ile)	\$ 39
256K/80 Column BD - w/Software (Ile)	\$109
80 Column BD - Videx Compatible (II+)	\$ 49
Z80 CP/M BD (II+/Ile)	\$ 38
16K Ram BD - w/Cable (II+)	\$ 35
128K Ram BD (II+/Ile)	\$ 89
Graphic Parallel BD - w/Cable (II+/Ile)	\$ 45
Super Serial BRd - w/cable	\$ 49
GS Super Cooler Fan	\$ 25
Disk Drive HH (Specify II+/Ile/ or Ilc)	\$129
Numeric Key Pad-16 Keys (Ile)	\$ 35
Coolin Fan w/Surge Protector (II+/Ile)	\$ 28
Joy Stick (Specify II+/Ile or Ilc)	\$ 15
Joystick w/Large Fire Button	\$ 25
Mini Vacuum Cleaner	\$ 10
Two Control SW Boxes (A/B)	
For Centronic/RS232 Connectors	\$ 29
RS232 Gender Changers M/M - F/F	\$ 5
RS232 Jumper Boxes - Disk Notchers	\$ 5
RS232 Data Testers	\$ 8

**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**

## TRAC CARD



### Boot Process Memory Card

- +On-Board Memory Stores Up To 200 Disks Of Accessed Tracks While Powerd Up
- +All Disks Are Automatically Monitored From The Moment You Power Up. The Tracks Are Divided Into Groups Of "Booted" Disks
- +Save Time When Using Backup Software-The Tracks Accessed May Be Displayed In Numerical Order Or In The Order In Which They Are Read
- +TRAC CARD Gives You Maximum Accuracy For Backing Up Software By Precisely Storing 1/4, 1/2 and 3/4 Tracks, As Well As Full Tracks
- +You May Choose 40 or 80 Column On Monitor Or Dump Data To Printer. Name Each Disk When Printing Track List
- +Choose Either Decimal Or Hexidecimal Readout
- +Use In Any Slot, Including Slot #3 On //e
- +Works With Any Apple Compatible 5 1/4" Drive
- +Works With Apple II, II+ and //e, As Well As Compatibles

Price \$159.95 Plus \$3.00 Shipping & Handling

Personal checks, M.O.,  
Visa and Mastercard  
Phone 913 676-7242

## TRAK STAR



### Constant Digital Readout of Disk Drive Head Position

- +Works With Any 5 1/4" Apple Compatible Drive
- +Saves Copying Time With Nibble Programs
- +Copy Only Tracks That Are Displayed
- +If Copied Program Doesn't Run, TRAK STAR Displays Track To Be Recopied
- +Displays Full and Half Tracks
- +Operates With Any Apple Compatible Program, Including Protected Software
- +Displays Up To 99 Tracks and Half Tracks; Compatible With High Density Drives
- +Does Not Use A Slot in the Apple
- +For Apple II, II+ and //e
- +Simple One Minute Installation

Price \$99.95 Plus \$3.00 Shipping & Handling  
Adaptor Cable Required For 2 Drive System \$12.00  
DuoDisk, 5 1/4" Unidisk and Ilc Owners Please Write

Apple is a registered trademark of Apple Computer Inc.

**Midwest Microsystems**

10308 Metcalf, Suite 355  
Overland Park, KS. 66212



# COMPUTIST

**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 and the Lower Case Option For The II Plus • Amazing Computer Facts • Shape Magic utility program • 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 the 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 • International Gran Prix • *Features* • Making DOSless Utilities • Pixit Printer Drivers • *Core* • Review of Z-RAM Memory Expansion Board • Reading the Joystick.....

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

**42** *Softkeys* • Light Simulator • *Readers' Softkeys* • Beach-Head • Monty Plays Scrabble • Racter • Winnie the Pooh • Infocom Stuff, Kabul Spy, Prisoner II • Wizardry 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 • *Readers' Softkeys* • Story Maker • Adventure Writer • Mummy's Curse • Zaxxon • The Quest • Pitfall II • H.E.R.O. • *Features* • A Two-Drive Patch for Winter Games • Customizing the Speed of a Duodisk • Roll the Presses Part Two: Printshop Printer Drivers • The Games of 1986.....

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

**39** *Softkeys* • MIDI/8 Plus • Homeword v2.1 • Borrowed Time • Amazon • Speed Reader II • *Readers' Softkeys* • 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 • *Hints* • Carmen Sandiego.....

**38** *Softkeys* • Cyclod • Alternate Realty • Boulder Dash I & II • Hard Hat Mack (Revisited) • The Other Side • *Readers' Softkeys* • F-15 Strike Eagle • Championship Lode Runner • Gato V 1.3 • I, Damiano • Wilderness • Golf's Best • *Features* • The Enhanced/Unenhanced II • 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) • *Readers' Softkeys* • Magic Slate • Alter Ego • Rendezvous • Quicken • Story Tree • Assembly Language Tutor • Avalon Hill games • Dark Crystal • *Features* • Playing Karateka on a //c • Track Finder • Sylk to Dif • *Core* • Breaking In: tips for beginners • Copy II Plus 6.0: a review • The DOS Alterer.....

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

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

**34** *Softkeys* • Crisis Mountain • Terripin Logo • Apple Logo II • Fishies 1.0 • SpellWorks • Gumball • *Readers' Softkeys* • 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 • *Readers' Softkeys* • Graphic Adventure #1 • Sensible Grammar & Extended Bookends • Chipwits • Hardball • King's Quest II • The World's Greatest Baseball Game • *Feature* • How to be the Sound Master • *Core* • The Mapping of Ultima IV.....

**32** *Softkeys* • Revisiting Music Construction Set • Cubit • Baudville Software • Hartley Software • Bridge • Early Games for Young Children • Tawala's Last Redoubt • *Readers' Softkeys* • Print Shop Companion • 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 • *Readers' Softkeys* • Time Zone • Tycoon • Earthly Delights • Jingle Disk • Crystal Caverns • Karate Champ • *Feature* • A Little Help With The Bard's Tale • *Core* • Black Box • Unrestricted Ampersand.....



# Available Back Issues

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

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

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

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

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

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

**24** *Softkeys* • Electronic Arts software • Grolier software • Xyphus • F-15 Strike Eagle • Injured Engine • *Readers' Softkeys* • Mr. Robot And His Robot Factory • 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 • Mufplot • Flashcalc • Karateka • Newsroom • E-Z Draw • *Readers' Softkeys* • Gato • Dino Eggs • Pinball Construction Set • TAC • The Print Shop: Graphics Library • Death In The Caribbean • *Features* • Using A.R.D. To Softkey Mars Cars • How To Be The Writemaster • *Core* • Wheel Of Money.....

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

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

**19** *Readers' 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 • *Readers' Softkeys* • Rescue Raiders • Sheila • Basic Building Blocks • Artsci Programs • Crossfire • *Feature* • Secret Weapon: RAMcard • *Core* • The Controller Writer • A Fix For The Beyond Castle Wolfenstein Softkey • The Lone Catalog Arranger Part 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

Use the handy  
Order Form  
on the back.

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

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

Issue	Mag \$4.75	Disk \$9.95	Both \$12.95
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"/>
20	<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"/>
Core 3	<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 \$10.00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
(All three CORE magazines)			

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

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

## 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 \_\_\_\_\_ CP47

US funds drawn on US banks

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.



# How To Write for **COMPUTIST**

COMPUTIST has two basic sections:  
(1) the new Readers Data Exchange, and  
(2) Special Features.

## **R D Ex** **Readers Data Exchange**

The old INPUT and the READERS SOFTKEY AND COPY EXCHANGE have now been combined into the new Readers Data Exchange (or RDEx). All short softkeys, comments, short articles, questions, answers, fixes and bugs and almost any other information that must be printed in a timely fashion will be placed in the Readers Data Exchange.

Although we will print letters and short article-softkey submissions that are received as only printed text, we would prefer that they be submitted on disk. For submission requirements, see *How To Get Published*.

We no longer purchase short softkeys-articles unless they merit being placed in the SPECIAL FEATURES section.

COMPUTIST reserves the right to edit and publish any contribution (whether a letter-to-the-editor or short article-softkey) in the RDEx section. All published contributions become the property and are copyrighted as part of COMPUTIST.

## **COMPUTIST** **SPECIAL FEATURES**

We will purchase **feature-length** original material on the following subjects:

- 1) Original program/article combinations;
- 2) General articles (Apple computing);
- 3) Softkeys;
- 4) Hardware modifications;
- 6) DOS modifications.

COMPUTIST pays upon publication, depending upon the amount of editing and typeset length of the article, from \$20 to \$100 per article.

Please submit completed manuscripts directly; do not send queries. Previously published material and simultaneous submissions will not be purchased.

## **HOW TO GET PUBLISHED**

Whether you are contributing data to the new Readers Data Exchange or if you feel your submission is worthy of being published as a Special Features, we urge you to follow these guidelines and please PLEASE send contributions and submissions on disk as well as on paper!

All contributions-submissions should be submitted in BOTH printed and disk formats. 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.

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.

Disks are always returned as soon as possible. Other materials will be returned only when adequate return packaging and postage is enclosed. We are not responsible for unreturned submissions.

## **HINTS ON WRITING STYLES**

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

B. 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 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. For each softkey, first introduce the locking technique used and then give precise steps to unlock the copy-protected program. Number each step. 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*. These will be **IMMEDIATELY** edited, typeset and entered into the Readers Data Exchange. Remember that we cannot purchase these contributions. Feature-length articles should be sent to the *Features Editor*. You will be notified of the status of your submission within 6 to 8 weeks after it is received.

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



# Looking for the Best Deal in Town?

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

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

### The Super IOB program.

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

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

### This package contains:

► TWO DISKS (supplied in DOS 3.3). Each disk contains at least 60 Super IOB Controllers including the standard, swap, newswap and fast controllers. Also included is **version 1.5 of Super IOB**, the 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 Caribbean, Dino Eggs, DLM Software, Electronic Arts, F-15 Strike Eagle, Fast Controller, Fathoms 40, Financial Cookbook, Gessler Software, Grandma's House, The Heist, In Search of the Most Amazing Thing, Instant Recall, Kidwriter, Lions Share, Lode Runner, Mastertype, Match Maker, Miner 2049er, Minit Man, Mufplot, Newsroom, Newswap controller, Penguin Software, Print Shop Graphic Library, Print Shop, Rendezvous with Rama, Rockys' Boots, Sargon III, Sea Dragon, Shiela, Skyfox, Snooper Troops, Standard controller, Stoneware Software, Summer Games, Super Controller, Super Zaxxon, Swap Controller, TAC, Ultima I II, Word Challenge, Xyphus, Zaxxon

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

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

Yes, please send me The Super IOB Collection

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

- US/Canada/Mexico for \$16 00  
 Other Foreign for \$20.00

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

\*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 \_\_\_\_\_

Address \_\_\_\_\_

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

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

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

Signature \_\_\_\_\_ CP47

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